A Comprehensive Survey on Local Differential Privacy toward Data Statistics and Analysis

Teng Wang [1,*][iD], Xuefeng Zhang [1][iD], Jingyu Feng [1] and Xinyu Yang [2]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; zhangxuefeng3@163.com (X.Z.); fengjy@xupt.edu.cn (J.F.)
[2] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China; yxyphd@mail.xjtu.edu.cn
[*] Correspondence: wangteng@xupt.edu.cn

check for updates

Abstract: Collecting and analyzing massive data generated from smart devices have become increasingly pervasive in crowdsensing, which are the building blocks for data-driven decision-making. However, extensive statistics and analysis of such data will seriously threaten the privacy of participating users. Local differential privacy (LDP) was proposed as an excellent and prevalent privacy model with distributed architecture, which can provide strong privacy guarantees for each user while collecting and analyzing data. LDP ensures that each user's data is locally perturbed first in the client-side and then sent to the server-side, thereby protecting data from privacy leaks on both the client-side and server-side. This survey presents a comprehensive and systematic overview of LDP with respect to privacy models, research tasks, enabling mechanisms, and various applications. Specifically, we first provide a theoretical summarization of LDP, including the LDP model, the variants of LDP, and the basic framework of LDP algorithms. Then, we investigate and compare the diverse LDP mechanisms for various data statistics and analysis tasks from the perspectives of frequency estimation, mean estimation, and machine learning. Furthermore, we also summarize practical LDP-based application scenarios. Finally, we outline several future research directions under LDP.

Keywords: local differential privacy; data statistics and analysis; enabling mechanisms; applications

1. Introduction

With the rapid development of wireless communication techniques, Internet-connected devices (e.g., smart devices and IoT appliances) are ever-increasing and generate large amounts of data by crowdsensing [1]. Undeniably, these big data have brought our rich knowledge and enormous benefits, which deeply facilitates our daily lives, such as traffic flow control, epidemic prediction, and recommendation systems [2–4]. To make better collective decisions and improve service quality, a variety of applications collect users data through crowdsensing to analyze statistical knowledge of the social community [5]. For example, the third-parties learn rating aggregation by gathering preference options [6], present a crowd density map by recording users locations [7], and estimate the power usage distributions from meter readings [8,9]. Almost all data statistics and analysis tasks fundamentally depend on a basic understanding of the distribution of the data.

However, collecting and analyzing data has incurred serious privacy issues since such data contain various sensitive information of users [10–12]. Even worse is that, driven by advanced data fusion and analysis techniques, the private data of users are more vulnerable to attack and disclosure in the big data era [13–15]. For example, the adversaries can infer the daily habits or behavior profiles of family members (e.g., the time of presence/absence in the home, certain activities such as watching TV,

cooking) by analyzing the usage of appliances [16–18], and even obtain = identification information, social relationships, and attitudes towards religion [19].

Therefore, it is an urgent priority to put great attention on preventing personal data from being leaked when collecting data from various devices. At present, the European Union (EU) has published the GDPR [20] that regulates the EU laws of data protection for all individual citizens and contains the provisions and requirements pertaining to the processing of personal data. Besides, the NIST of the U.S. is also developing the privacy frameworks [21] currently to better identify, access, manage, and communicate about privacy risks so that individuals can enjoy the benefits of innovative technologies with greater confidence and trust.

From the perspective of privacy-preserving techniques, differential privacy (DP) [22] was proposed for more than ten years and recognized as a convincing framework for privacy protection, which also refers to global DP (or centralized DP). (Without loss of generality, DP appears in the rest of this article refers to global DP (i.e., centralized DP).) With strict mathematical proofs, DP is independent of the background knowledge of adversaries and capable of providing each user with strong privacy guarantees, which was widely adopted and used in many areas [23,24]. However, DP can be only used to the assumption of a trusted server. In many online services or crowdsourcing systems, the servers are untrustworthy and always interested in the statistics of users' data.

Based on the definition of DP, local differential privacy (LDP) [25] is proposed as a distributed variant of DP, which achieves privacy guarantees for each user locally and is independent of any assumptions on the third-party servers. LDP was imposed as the cutting-edge of research on privacy protection and risen in prominence not only from theoretical interests, but also subsequently from a practical perspective. For example, many companies deployed LDP-based algorithms in real systems, such as Apple iOS [26], Google Chrome [27], Windows system [28].

Due to its powerfulness, LDP was widely adopted to alleviate the privacy concerns of each user while conducting statistical and analytic tasks, such as frequency and mean value estimation [29], heavy hitters discovery [30], *k*-way marginal release [31], empirical risk minimization (ERM) [32], federated learning [33], and deep learning [34].

Therefore, a comprehensive survey of LDP is very necessary and urgent for future research in Internet of Things. To the best of our knowledge, only a little literature focuses on reviewing LDP and the most existing surveys only pay attention to a certain field. For example, Wang et al. [35] summarized several LDP protocols only for frequency estimation. The tutorials in [36–38] reviewed the LDP models and introduced the current research landscapes under LDP, but the detailed descriptions are rather insufficient. Zhao et al. [39] reviewed the existing LDP-based mechanisms only towards the Internet of connected vehicles. The reviews in [40,41] also provided a survey of statistical query and private learning with LDP. However, the detailed technical points and specific data types when using LDP are still insufficiently summarized. Therefore, it is still necessary and urgent to carry out a comprehensive survey on LDP toward data statistics and analysis to help newcomers understand the complex discipline of this hot research area.

In this survey, we conduct an in-depth overview of LDP with respect to its privacy models, the related research tasks for various data, enabling mechanisms, and wide applications. Our main contributions are summarized as follows.

1. We firstly provide a theoretical summarization of LDP from the perspectives of the LDP models, the general framework of LDP algorithms, and the variants of LDP.
2. We systematically investigate and summarize the enabling LDP mechanisms for various data statistics and analysis tasks. In particular, the existing state-of-the-art LDP mechanisms are thoroughly concluded from the perspectives of frequency estimation, mean value estimation, and machine learning.
3. We explore the practical applications with LDP to show how LDP is to be implemented in various applications, including in real systems (e.g., Google Chrome, Apple iOS), edge computing, hypothesis testing, social networks, and recommendation systems.

4.   We further distinguish some promising research directions of LDP, which can provide useful guidance for new researchers.

Figure 1 presents the main research categories of LDP and also shows the main structure of this survey. We first provide a theoretical summarization of LDP, which includes the LDP model, the framework of LDP algorithms, and the variants of LDP. Then, from the perspective research tasks, we summarize the existing LDP-based privacy-preserving mechanisms into three categories: frequency estimation , mean estimation and machine learning. We further subdivide each category into several subtasks based on different data types. In addition, we summarize the applications of LDP in real practice and other fields.
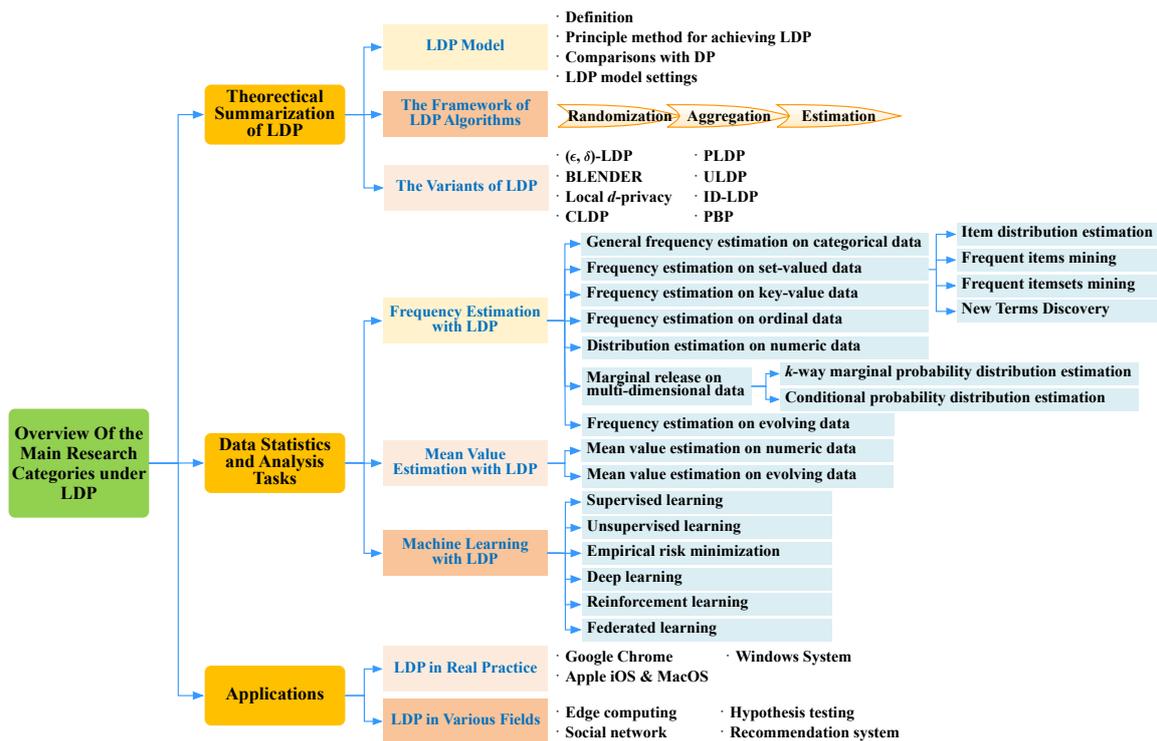

Figure 1. An overview of the main research categories with LDP.

The rest paper is organized as follows. Section 2 theoretically summarizes the LDP. The diverse LDP mechanisms for frequency estimation, mean estimation and machine learning are introduced thoroughly in Sections 3–5, respectively. Section 6 summarizes the wide application scenarios of LDP and Section 7 presents some future research directions. Finally, we conclude the paper in Section 8.

2. Theoretical Summarization of LDP

Formally, let N be the number of users, and $U_i(1 \leq i \leq N)$ denote the i-th user. Let V^i denote the data record of U_i, which is sampled from the attribute domain $\mathcal{A}$ that consists of d attributes $A_1, A_2, \cdots, A_d$. For categorical attribute, its discrete domain is denoted as $\mathcal{K} = \{v_1, v_2, \cdots, v_k\}$, where k is the size of the domain and $|\mathcal{K}| = k$. Notations commonly used in this paper are listed in Table 1.

Table 1. The commonly used notations.

Notation	Explanation
V^i	Data record of user U_i
$\mathcal{K} = \{v_1, \cdots, v_k\}$	Domain of categorical attribute with size $|\mathcal{K}| = k$
v/v^*	Input value/Perturbed value
B	Vector of the encoded value
N	Number of users
$N_v/\bar{N}_v/\widehat{N}_v$	The true/reported/estimated number of value v
A	Attribute
d	Dimension
ϵ/δ	Privacy budget/Probability of failure
p, q	Perturbation probability
$f_v/\widehat{f}_v$	The true/estimated frequency of value v
$\mathbb{H}/H$	Hash function universe/Hash function

2.1. LDP Model

Local differential privacy is a distributed variant of DP. It allows each user to report her/his value v locally and send the perturbed data to the server aggregator. Therefore, the aggregator will never access to the true data of each user, thus providing a strong protection. Here, user's value v acts as the input value of a perturbation mechanism and the perturbed data acts as the output value.

2.1.1. Definition

Definition 1 (**ϵ-Local Differential Privacy** ([ϵ-LDP) [25,42]). *A randomized mechanism $\mathcal{M}$ satisfies ϵ-LDP if and only if for any pairs of input values v, v' in the domain of $\mathcal{M}$, and for any possible output $y \in \mathcal{Y}$, it holds*

$$\mathbb{P}[\mathcal{M}(v) = y] \leq e^\epsilon \cdot \mathbb{P}[\mathcal{M}(v') = y], \tag{1}$$

where $\mathbb{P}[\cdot]$ denotes probability and ϵ is the privacy budget. A smaller ϵ means stronger privacy protection, and vice versa.

Sequential composition is a key theorem of LDP, which plays important roles in some complex LDP algorithms or some complex scenarios.

Theorem 1 (**Sequential Composition**). *Let $\mathcal{M}_i(v)$ be an ϵ_i-LDP algorithm on an input value v, and $\mathcal{M}(v)$ is the sequential composition of $\mathcal{M}_1(v), ..., \mathcal{M}_m(v)$. Then $\mathcal{M}(v)$ satisfies $\sum_{i=1}^m \epsilon_i$-LDP.*

2.1.2. The Principle Method for Achieving LDP

Randomized response (RR) [43] is the classical technique for achieving LDP, which can also be used for achieving global DP [44]. The main idea of RR is to protect user's private information by answering a plausible response to the sensitive query. That is, one user who possesses a private bit x flips it with probability p to give the true answer and with probability $1 - p$ to give other answers.

For example, the data collector wants to count the true proportion f of the smoker among N users. Each user is required to answer the question "Are you a smoker?" with "Yes" or "No". To protect privacy, each user flips an unfair coin with the probability p being head and the probability $1 - p$ being tail. If the coin comes up to the head, the user will respond the true answer. Otherwise, the user will respond the opposite answer. In this way, the probabilities of answering "Yes" and "No" can be calculated as

$$\mathbb{P}[\text{answer=``Yes''}] = fp + (1 - f)(1 - p), \tag{2}$$

$$\mathbb{P}[\text{answer=``No''}] = (1 - f)p + f(1 - p). \tag{3}$$

Then, we estimate the number of "Yes" and "No" that are denoted as N_1 and $N - N_1$. From Equations (2) and (3), we have $N_1/N = fp + (1-f)(1-p)$ and $(N-N_1)/N = (1-f)p + f(1-p)$. Therefore, we can compute the estimated proportion of the smoker is

$$\widehat{f} = \frac{p-1}{2p-1} + \frac{N_1}{(2p-1)N}.$$
(4)

Observe that in the above example the probability of receiving "Yes" varies from p to $1-p$ depending on the true information of users. Similarly, the probability of receiving "No" also varies from p to $1-p$. Hence, the ratio of probabilities for different answers of one user can be at most $\frac{p}{1-p}$. By letting $\frac{p}{1-p} = e^\epsilon$ and based on Equation (1), it can be easily verified that the above example satisfies $(\ln \frac{p}{1-p})$-LDP. To ensure $(\ln \frac{p}{1-p}) > 0$, we should make sure that $p > \frac{1}{2}$.

Therefore, RR achieves LDP by providing plausible deniability for the responses of users. In this case, users no longer need to trust a centralized curator since they only report plausible data. Based on RR, there are plenty of other mechanisms for achieving LDP under different research tasks, which will be introduced in the following Sections.

2.1.3. Comparisons with Global Differential Privacy

We compare LDP with global DP from different perspectives, as shown in Table 2. At first, the biggest difference between LDP and DP is that LDP is a local privacy model with no assumption on the server while DP is a central privacy model with the assumption of a trusted server. Correspondingly, the general processing frameworks of DP and LDP are different. As shown in the left part of Figure 2, under the DP framework, the data are directly sent to the server and the noises are added to query mechanisms in the server-side. In contrast, under the LDP framework, each user's data are locally perturbed in the client-side before uploading to the server, as shown in the right part of Figure 2.
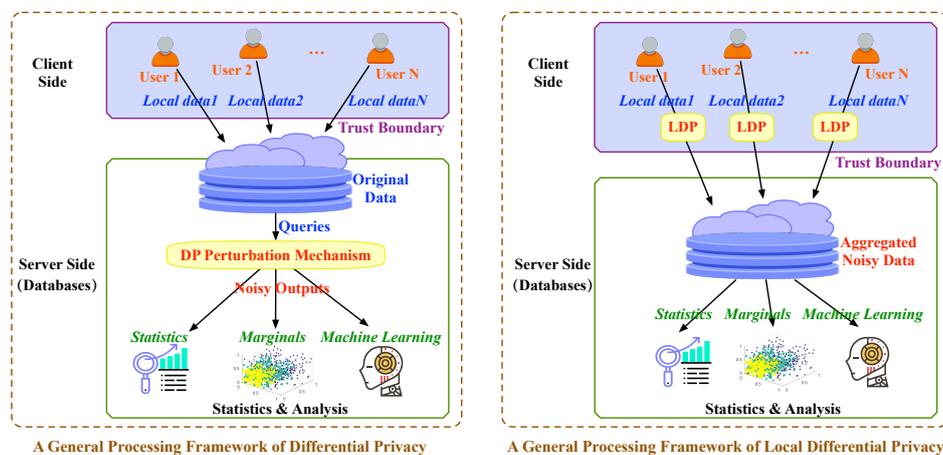

Figure 2. The general processing frameworks of DP and LDP.

Table 2. Comparisons between LDP and DP.

Notion	Model	Server	Neighboring Datasets	Basic Mechanism	Property	Applications
DP [22,45]	Central	Trusted	Two datasets	Laplace/Exponential Mechanisms [45,46]	Sequential Composition, Post-processing	Data collection, statistics, publishing, analysis
LDP [25,42]	Local	No requirement	Two records	Randomized Response [42,43]		

The neighboring datasets in LDP are defined as two different records/values of the input domain. While in DP, the neighboring datasets are defined as two datasets that differ only in one record. For example, given a dataset, we can get its neighboring dataset by deleting/modifying one record. The most two common perturbation mechanisms of achieving DP are Laplace mechanism

and Exponential mechanism [45,46] that inject random noises based on privacy budget and sensitivity. In contrast, the randomized response technique [42,43] is most commonly used to achieve LDP. As shown in Table 2, LDP holds the same sequential composition and post-processing properties as DP. Both DP and LDP are widely adopted by many applications, such as data collection, publishing, analysis, and so on.

2.1.4. LDP Model Settings

This section summarizes the model settings of LDP, which holds two paradigms, i.e., interactive setting and non-interactive setting [42,47].

Since LDP no longer assumes a trusted third-party data curator, the interactive and non-interactive privacy model settings of LDP [47] are different from that of DP [23]. Figure 3 shows the interactive and non-interactive settings of LDP.

Let $v_1, v_2, \cdots, v_n \in \mathcal{K}$ be the input sequences, and $y_1, y_2, \cdots, y_n \in \mathcal{Y}$ be the corresponding output sequences. As shown in left part of Figure 3, in interactive setting, the i-th output y_i depends on the i-th input v_i and the previous $i-1$ outputs $y_{1:i-1}$, but is independent of the previous $i-1$ inputs $v_{1:i-1}$. Particularly, the dependence and conditional independence correlations can be formally denoted as $\{v_i, y_1, \cdots, y_{i-1}\} \rightarrow y_i \wedge y_i \perp v_j | \{v_i, y_1, \cdots, y_{i-1}\}$ for any $j \neq i$.

In contrast, as shown in the right part of Figure 3, the non-interactive setting is much simpler than interactive setting. The i-th output y_i only depends on the i-th input v_i. In formal, the dependence and conditional independence correlations can be denoted as $v_i \rightarrow y_i \wedge y_i \perp \{v_j, y_j, j \neq i\} | v_i$.

Therefore, the main difference between interactive and non-interactive settings of LDP is whether to consider the correlations between the output results. The work in [48,49] further investigated the power of interactivity in LDP.
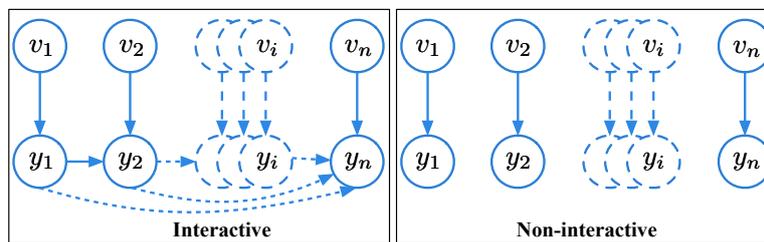

Figure 3. LDP model settings.

2.2. The Framework of LDP Algorithm

The general privacy-preserving framework with LDP includes three modules: Randomization, Aggregation and Estimation, as shown in Algorithm 1. The randomization is conducted in the client side and both aggregation and estimation happen in the server side.

Algorithm 1: The General Procedure of LDP-based Privacy-preserving Mechanisms

---

Input: original data of the users, privacy parameter ϵ
Output: objective information f
   /* Randomization in the client side   */
1 for *each user U_i $(i \in [1, N])$ with value v* **do**
2   Encode input value when necessary;
3   Conduct perturbation with privacy parameter ϵ;
4   Report noisy value $\widehat{v}$ to the server;

   /* Aggregation in the server side   */
5 Aggregate all reported values;
   /* Estimation in the server side   */
6 Estimate and correct the objective information f;
7 return f

2.3. The Variants of LDP

Since the introduction of LDP, designing the variant of LDP was an important research direction to improve the utility of LDP and to make LDP more relevant in targeted IoT scenarios. This section summarizes the current research progresses on LDP variant, as shown in Table 3.

2.3.1. (ϵ, δ)-LDP

Similar to the case that (ϵ, δ)-DP [50] is a relaxation of ϵ-DP, (ϵ, δ)-LDP (also called approximate LDP) is a relaxation of ϵ-LDP (also called pure LDP).

Definition 2 (**[(ϵ, δ)-Local Differential Privacy ([(ϵ, δ)-LDP) [51]**]). *A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-LDP if and only if for any pairs of input values v and v' in the domain of $\mathcal{M}$, and for any possible output $y \in \mathcal{Y}$, it holds*

$$\mathbb{P}[\mathcal{M}(v) = y] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(v') = y] + \delta, \tag{5}$$

where δ is typically small.

Loosely speaking, (ϵ, δ)-LDP means that a mechanism $\mathcal{M}$ achieves ϵ-LDP with probability at least $1 - \delta$. By relaxing ϵ-LDP, (ϵ, δ)-LDP is more general since the latter in the special case of $\delta = 0$ becomes the former.

2.3.2. BLENDER

BLENDER [52] is a hybrid model by combining global DP and LDP, which improves data utility with desired privacy guarantees. The BLENDER is achieved by separating the user pool into two groups based on their trust in the data aggregator. One is called opt-in group that contains the users who have higher trust in the aggregator. Another is called clients that contains the remaining users. Then, the BLENDER can maximize the data utility by balancing the data obtained from participation of opt-in users with that of other users. The privacy definition of BLENDER is the same as (ϵ, δ)-DP [50].

2.3.3. Local d-Privacy

Geo-indistinguishability [53] was initially proposed for location privacy protection under global DP, which is defined based on the geographical distance of data. Geo-indistinguishability was quite successful when the statistics are distance-sensitive. In the local settings, Alvim et al. [54] also pointed out that the metric-based LDP can provide better utility that standard LDP. Therefore, based on d-privacy [55], Alvim et al. [54] proposed local d-privacy that is as defined as follows.

Definition 3 (**Local d-Privacy**). *A randomized mechanism $\mathcal{M}$ satisfies local d-privacy if and only if for any pairs of input values v and v' in the domain of $\mathcal{M}$, and for any possible output $y \in \mathcal{Y}$, it holds*

$$\mathbb{P}[\mathcal{M}(v) = y] \leq e^{\epsilon \cdot d(v,v')} \cdot \mathbb{P}[\mathcal{M}(v') = y], \tag{6}$$

where $d(\cdot, \cdot)$ is a distance metric.

Local d-privacy can relax the privacy constraint by introducing a distance metric when $d(v, v') > 1$, thus improving data utility. In other words, the relaxation of local d-privacy is reflected in that the two data becomes more distinguishable as their distance increases. Therefore, local d-privacy is quite appropriate for distance-sensitive data, such as location data, energy consumption in smart meters.

2.3.4. CLDP

LDP played an important role in data statistics and analysis. However, the standard LDP will suffer a poorly data utility when the number of users is small. To address this, Gursoy et al. [56] introduced condensed local differential privacy (CLDP) that is also a metric-based privacy notation. Let $d(\cdot, \cdot)$ be a distance metric. Then, CLDP is defined as follows.

Definition 4 (α-CLDP). *A randomized mechanism $\mathcal{M}$ satisfies α-CLDP if and only if for any pairs of input values v and v' in the domain of $\mathcal{M}$, and for any possible output $y \in \mathcal{Y}$, it holds*

$$\mathbb{P}[\mathcal{M}(v) = y] \le e^{\alpha \cdot d(v,v')} \cdot \mathbb{P}[\mathcal{M}(v') = y], \tag{7}$$

where $\alpha > 0$.

By definition, in CLDP, α must decrease to compensate as distance d increases. Thus, it holds that $\alpha \ll \epsilon$. In addition, Gursoy et al. [56] also adopted a variant of the Exponential Mechanism (EM) to design several protocols that achieve CLDP with better data utility when there is a small number of users.

2.3.5. PLDP

Instead setting a global privacy constraint for all users, personalized local differential privacy (PLDP) [7,57] is proposed to provide granular privacy constraints for each participating user. That is, under PLDP, each user can select the privacy demand (i.e., ϵ) according to his/her own preference. In formal, PLDP is defined as follows.

Definition 5 (ϵ-PLDP). *A randomized mechanism $\mathcal{M}$ satisfies ϵ_U-PLDP if and only if for any pairs of input values v and v' in the domain of $\mathcal{M}$ and a user U, and for any possible output $y \in \mathcal{Y}$, it holds*

$$\mathbb{P}[\mathcal{M}(v) = y] \le e^{\epsilon_U} \cdot \mathbb{P}[\mathcal{M}(v') = y], \tag{8}$$

where ϵ_U is the privacy budget belonging to user U.

To achieve PLDP, Chen et al. [7] proposed personalized count estimation (PCE) protocol and further leveraged a user group clustering algorithm to apply PCE to users with different privacy level. In addition, Nie et al. [57] proposed the advanced combination strategy to compose multilevel privacy demand with an optimal utility.

2.3.6. ULDP

The standard LDP regards all user data equally sensitive, which leads to excessive perturbations. In fact, not all personal data are equally sensitive. For example, answer a questionnaires such as: "Are you a smoker?" Obviously, "Yes" is a sensitive answer, whereas "No" is not sensitive. To improve data utility, Utility-optimized LDP (ULDP) [58] was proposed as a new privacy notation to provide privacy guarantees only for sensitive data. In ULDP, let $\mathcal{K}_S \subseteq \mathcal{K}$ be the sensitive data set, and $\mathcal{K}_N = \mathcal{K} \setminus \mathcal{K}_S$ be the remaining data set. Let $\mathcal{Y}_P \subseteq \mathcal{Y}$ be the protected data set, and $\mathcal{Y}_I = \mathcal{Y} \setminus \mathcal{Y}_P$ be the invertible data set. Then, ULDP is formally defined as follows.

Definition 6 (($\mathcal{K}_S, \mathcal{Y}_P, \epsilon$)-ULDP). *Given $\mathcal{K}_S \subseteq \mathcal{K}$, $\mathcal{Y}_P \subseteq \mathcal{Y}$, a randomized mechanism $\mathcal{M}$ provides ($\mathcal{K}_S, \mathcal{Y}_P, \epsilon$)-PLDP if it satisfies the following properties:*
(i) For any $y \in \mathcal{Y}_I$, there exists an $v \in \mathcal{X}_N$ such that

$$\mathbb{P}[\mathcal{M}(v) = y] > 0, \mathbb{P}[\mathcal{M}(v') = y] = 0 \text{ for any } x' \ne x \tag{9}$$

(ii) For any $v, v' \in \mathcal{K}$ and any $y \in \mathcal{Y}_P$,

$$\mathbb{P}[\mathcal{M}(v) = y] \le e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(v') = y] \tag{10}$$

For an intuitive understanding for Definition 6, $(\mathcal{K}_S, \mathcal{Y}_P, \epsilon)$-ULDP maps sensitive data $v \in \mathcal{K}_S$ to only protected data set. Specifically, we can see from Formula (9) that no privacy protects are provided for non-sensitive data since each output in $\mathcal{Y}_I$ reveals the corresponding input in $\mathcal{K}_N$. Also, we can also find from Formula (10) that $(\mathcal{K}_S, \mathcal{Y}_P, \epsilon)$-ULDP provides the same privacy protections as ϵ-LDP for all sensitive data $v \in \mathcal{K}_S$.

2.3.7. ID-LDP

In ULDP, Murakami et al. [58] considered the sensitivity level of input data by directly separating the input data into sensitive data and non-sensitive data. However, Gu et al. [59] further indicated that different data have distinct sensitivity levels. Thus, they presented the Input-Discriminative LDP (ID-LDP) which is a more fine-grained version of LDP. The notion of ID-LDP is defined as follows.

Definition 7 ($\mathcal{E}$-ID-LDP). *For a given privacy budget set $\mathcal{E} = \{\epsilon_v\}_{v \in \mathcal{K}}$, a randomized mechanism $\mathcal{M}$ satisfies $\mathcal{E}$-ID-LDP if and only if for any pairs of input values v and v', and for any possible output $y \in \mathcal{Y}$, it holds*

$$\mathbb{P}[\mathcal{M}(v) = y] \le e^{r(\epsilon_v, \epsilon_{v'})} \cdot \mathbb{P}[\mathcal{M}(v') = y] \tag{11}$$

where $r(\cdot, \cdot)$ is a function of two privacy budget.

It can be seen from Definition 7 that ID-LDP introduces the function $r(\epsilon_v, \epsilon_{v'})$ to quantify the indistinguishability between input values v and v' that have different privacy levels with privacy budget ϵ_v and $\epsilon v'$. The work in [58] mainly considers the minimum function between ϵ_v and $\epsilon v'$ and formalizes the MinID-LDP as follows.

Definition 8 (MinID-LDP). *A randomized mechanism $\mathcal{M}$ satisfies $\mathcal{E}$-MinID-LDP if and only if it satisfies $\mathcal{E}$-ID-LDP with $r(\epsilon_v, \epsilon_{v'}) = \min\{\epsilon_v, \epsilon_{v'}\}$.*

That is, MinID-LDP always guarantees the worse-case privacy for the pair. Thus, MinID-LDP ensures better data utility by providing distinct protection for different inputs than standard LDP that provides the worse-case privacy for all data.

2.3.8. PBP

In addition, Takagi et al. [60] pointed that data providers can naturally choose and keep their privacy parameters secret since LDP perturbations occur in device side. Thus, they proposed a new privacy model Parameter Blending Privacy (PBP) as a generalization of standard LDP. PBP can not only keep the privacy parameters secret, but only improves the data utility through privacy amplification.

Let Θ be the domain of the privacy parameter. Given a privacy budget $\theta \in \Theta$, let $\mathbb{P}(\theta)$ be the ratio of the number of times that θ is chosen to the number of users. Then, PBP is defined as follows.

Definition 9 (r-PBP). *A randomized mechanism $\mathcal{M}$ satisfies r-PBP iff $\forall \theta \in \Theta, v, v' \in \mathcal{K}, y \in \mathcal{Y}, \exists \theta' \in \Theta$, it holds*

$$\mathbb{P}(\theta)\mathbb{P}[\mathcal{M}(v; \theta) = y] \le e^{r(\theta)} \cdot \mathbb{P}(\theta')\mathbb{P}[\mathcal{M}(v'; \theta') = y] \tag{12}$$

where the privacy function $r()$ returns a real number that denotes the strength of privacy protection.

Comparisons and discussions. Table 3 briefly summaries the various LDP variants from different perspectives. With various purposes, these variants extend the standard LDP into more generalized or granular versions based on different design ideas. Meanwhile, the main protocols for achieving such LDP variants are also proposed. Nonetheless, there are still some issues in new privacy notions remaining unsolved. For example, PBP only focuses on the privacy parameters that are chosen at the user-level. In other words, the correlations between data and privacy parameters are neglected in PBP. Similarly, ULDP can't be directly applied to scenarios that sensitive data and non-sensitive are correlated. Besides, MinID-LDP considers the minimum function to decide the privacy budget. There might be other functions that can provide better data utility.

Table 3. Summary of LDP variants (LDP is also listed for reference).

LDP Variants	Definition	Purpose	Design Idea	Target Data Type	Main Protocol	= LDP?
LDP [35]	$\frac{\mathbb{P}[\mathcal{M}(v)=y]}{\mathbb{P}[\mathcal{M}(v')=y]} \leq e^{\epsilon}$	-	-	All data type	RR-based method	-
(ϵ,δ)-LDP [61,62]	See Formula (5)	A relaxed variant of LDP	LDP fails with a small probability δ	All data type	RR-based method	When $\delta = 0$
BLENDER [52]	same as (ϵ,δ)-DP	Improve data utility by combine global DP and LDP	Group user pool	Categorical data	Laplace mechanism	-
Local d-privacy [54]	$\frac{\mathbb{P}[\mathcal{M}(v)=y]}{\mathbb{P}[\mathcal{M}(v')=y]} \leq e^{\epsilon \cdot \mathbf{d}(v,v')}$	Enhance data utility for metric spaces	Metric-based method	Metric data, e.g., location data	Discrete Laplace Geometric mechanisms	-
CLDP [56]	$\frac{\mathbb{P}[\mathcal{M}(v)=y]}{\mathbb{P}[\mathcal{M}(v')=y]} \leq e^{\alpha \cdot \mathbf{d}(v,v')}$	Solve the problem of a small number of users	Metric-based method	Categorical data	Exponential mechanism	-
PLDP [57]	$\frac{\mathbb{P}[\mathcal{M}(v)=y]}{\mathbb{P}[\mathcal{M}(v')=y]} \leq e^{\epsilon_U}$	Achieve granular privacy constraints	Advanced combination [57] PCE [7]	Categorical data	RR-based method	When $\epsilon_U = \epsilon$
ULDP [58]	See Definition 6	Optimize data utility	Only provide privacy guarantees for sensitive data	Categorical data	RR-based method	When $\mathcal{K}_S = \mathcal{K}$ and $\mathcal{Y}_P = \mathcal{Y}$
ID-LDP [59]	$\frac{\mathbb{P}[\mathcal{M}(v)=y]}{\mathbb{P}[\mathcal{M}(v')=y]} \leq e^{\mathbf{r}(\epsilon_x,\epsilon_{x'})}$	Provide input-discriminative protection for different inputs	Quantify indistinguishability	Categorical data	Unary Encoding	When $\epsilon_v = \epsilon$ for each value v
PBP [60]	See Definition 9	Achieve privacy amplification of LDP	Keep privacy parameters secret	Categorical data	RR-based method	-

3. Frequency Estimation with LDP

This section summarizes the state-of-the-art LDP algorithms for frequency estimation. Frequency estimation, which is equivalent to histogram estimation, aims at computing the frequency of each given value $v \in \mathcal{K}$, where $|\mathcal{K}| = k$. Besides, we further subdivide the frequency-based task under LDP into several more specific tasks. In what follows, we will introduce each LDP protocol in the view of randomization, aggregation, and estimation, as described in Section 2.2.

Based on the Definition 1 of LDP, a more visual definition of LDP protocol [35] can be given as follows.

Definition 10 (ϵ-LDP Protocol). *Consider two probabilities $p > q$. A local protocol given by $\mathcal{M}$ such that a user reports the true value with p and reports each of other values with q, will satisfy ϵ-LDP if and only if it holds $p \leq q \cdot e^{\epsilon}$.*

Based on the Theorem 2 in [35], the variance for the noisy number of the value v (i.e., $\widehat{N}_v$) among N users will be $\mathrm{Var}[\widehat{N}_v] = \frac{Nq(1-q)}{(p-q)^2} + \frac{N\widehat{f}_v(1-p-q)}{p-q}$, where f_v is the frequency of the value $v \in \mathcal{K}$. Thus, the variance is

$$\mathrm{Var}[\widehat{f}_v] = \frac{q(1-q)}{N(p-q)^2} + \frac{\widehat{f}_v(1-p-q)}{N(p-q)}. \tag{13}$$

It can be seen that the variance of Equation (13) will be dominated by the first term when f_v is small. Hence, the approximation of the variance in Equation (13) can be denoted as

$$\text{Var}^*[\widehat{f_v}] = \frac{q(1-q)}{N(p-q)^2}. \tag{14}$$

In addition, it holds that $\text{Var}^* = \text{Var}$ when $p + q = 1$.

3.1. General Frequency Estimation on Categorical Data

This section summarizes the general LDP protocols for frequency estimation on categorical data and shows the performance of each protocol. The encoding principle of the existing LDP protocols can be concluded as direct perturbation, unary encoding, hash encoding, transformation, and subset selection.

3.1.1. Direct Perturbation

The most basic building block for achieving LDP is direct perturbation that perturbs data directly by randomization.

Binary Randomized Response (BRR) [43,63] is the basic randomized response technique that focuses on binary values, i.e., the cardinality of value domain is 2. Section 2.1.2 introduced the basic randomized response technique that focuses on binary values. Based on this, BRR is formally defined as follows.

Randomization. Each value v is perturbed by

$$\mathbb{P}\left[\mathcal{M}(v) = v^*\right] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + 1}, & \text{if } v^* = v, \\ q = \frac{1}{e^\epsilon + 1}, & \text{if } v^* \neq v. \end{cases} \tag{15}$$

Aggregation and Estimation. Let $\widehat{N_v}$ be the total number of received value v after aggregation. The estimated frequency $\widehat{f_v}$ of value v can be computed as $\widehat{f_v} = \left(\frac{\widehat{N_v}}{N} - \frac{1}{e^\epsilon + 1}\right) \cdot \frac{e^\epsilon + 1}{e^\epsilon - 1}$.

Observe that the probability that $v^* = v$ varies from $\frac{1}{e^\epsilon + 1}$ to $\frac{e^\epsilon}{e^\epsilon + 1}$. The ratio of the respective probabilities for different values of v will be at most e^ϵ. Therefore, BRR satisfies ϵ-LDP. Based on Equation (14), the variance of BRR is

$$\text{Var}^*_{BRR}[\widehat{f_v}] = \frac{e^\epsilon}{N(e^\epsilon - 1)^2}. \tag{16}$$

Generalized Randomized Response (GRR) [64,65] extends the BRR to the case where the cardinality of total values is more than 2, i.e., $k > 2$. GRR is also called Direct Encoding (DE) in [35] or k-RR in [64]. The process of GRR is given as follows.

Randomization. Each value v is perturbed by

$$\mathbb{P}\left[\mathcal{M}(v) = v^*\right] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + k - 1}, & \text{if } v^* = v, \\ q = \frac{1}{e^\epsilon + k - 1}, & \text{if } v^* \neq v. \end{cases} \tag{17}$$

Aggregation and Estimation. Let $\widehat{N_v}$ be the total number of received value v after aggregation. The estimated frequency $\widehat{f_v}$ of value v can be computed as

$$\widehat{f_v} = \left(\frac{\widehat{N_v}}{N} - \frac{1}{e^\epsilon + k - 1}\right) \cdot \frac{e^\epsilon + k - 1}{e^\epsilon - 1}. \tag{18}$$

Observe that the probability that $v^* = v$ varies from $\frac{1}{e^\epsilon + k - 1}$ to $\frac{e^\epsilon}{e^\epsilon + k - 1}$. The ratio of the respective probabilities for different values of v will be at most e^ϵ. Therefore, GRR satisfies ϵ-LDP. Based on Equation (14), the variance of GRR is

$$\text{Var}^*_{GRR}[\widehat{f}_v] = \frac{(e^\epsilon + k - 2)}{N(e^\epsilon - 1)^2}. \tag{19}$$

3.1.2. Unary Encoding

Instead of perturbing the original value, we can perturb each bit of a vector that is generated by encoding the original value v. This method is called Unary Encoding (UE) [35] that is achieved as follows.

Randomization. UE encodes each value $v \in \mathcal{K}$ into a binary bit vector B with size k, where the v-th bit is 1, i.e., $B = [0, \cdots, 0, 1, 0, \cdots, 0]$. Each bit of B is perturbed by

$$\mathbb{P}\left[B^*[i] = 1\right] = \begin{cases} p, & \text{if } B[i] = 1, \\ q, & \text{if } B[i] = 0, \end{cases} \tag{20}$$

where $p > q$.

Aggregation and Estimation. Assume the number of ones in the v-th bit among all original N vectors and all received N vectors are N_v^1 and $\bar{N}_v^1$, respectively. Based on Equation (20), we have $\bar{N}_v^1 = N_v^1 p + (N - N_v^1)q$. Thus, the estimated number of value v is $\widehat{N}_v^1 = \frac{\bar{N}_v^1 - Nq}{p - q}$. Then, the frequency f_v of the value v is computed as

$$\widehat{f}_v = \frac{\widehat{N}_v^1}{N} = \left(\frac{\bar{N}_v^1}{N} - q\right) / (p - q). \tag{21}$$

Based on Equation (20), for any inputs $v_1 \in \mathcal{K}$ and $v_2 \in \mathcal{K}$, and the output B^*, it holds that

$$\begin{aligned}
\frac{\mathbb{P}\left[B^*|v_1\right]}{\mathbb{P}\left[B^*|v_2\right]} &= \frac{\prod_{i \in [k]} \mathbb{P}\left[B^*[i]|v_1\right]}{\prod_{i \in [k]} \mathbb{P}\left[B^*[i]|v_2\right]} \\
&\leq \frac{\mathbb{P}[B^*[v_1] = 1|v_1]\mathbb{P}[B^*[v_2] = 0|v_1]}{\mathbb{P}[B^*[v_1] = 1|v_2]\mathbb{P}[B^*[v_2] = 0|v_2]}
\end{aligned} \tag{22}$$

where "$\leq$" is achieved since the bit vectors differ only in positions v_1 and v_2. There are four cases when choosing values for positions v_1 and v_2. That is,

$$\begin{cases}
① \frac{\mathbb{P}[B^*[v_1] = 0|v_1] \cdot \mathbb{P}[B^*[v_2] = 0|v_1]}{\mathbb{P}[B^*[v_1] = 0|v_2] \cdot \mathbb{P}[B^*[v_2] = 0|v_2]} \\
② \frac{\mathbb{P}[B^*[v_1] = 0|v_1] \cdot \mathbb{P}[B^*[v_2] = 1|v_1]}{\mathbb{P}[B^*[v_1] = 0|v_2] \cdot \mathbb{P}[B^*[v_2] = 1|v_2]} \\
③ \frac{\mathbb{P}[B^*[v_1] = 1|v_1] \cdot \mathbb{P}[B^*[v_2] = 0|v_1]}{\mathbb{P}[B^*[v_1] = 1|v_2] \cdot \mathbb{P}[B^*[v_2] = 0|v_2]} \\
④ \frac{\mathbb{P}[B^*[v_1] = 1|v_1] \cdot \mathbb{P}[B^*[v_2] = 1|v_1]}{\mathbb{P}[B^*[v_1] = 1|v_2] \cdot \mathbb{P}[B^*[v_2] = 1|v_2]}
\end{cases} \tag{23}$$

It can be verified that a vector with position v_1 being 1 and position v_2 being 0 will maximize the ratio (i.e., the case ③).

Based on Equation (20), UE satisfies ϵ-LDP if and only if it follows that

$$\frac{\mathbb{P}\left[B^*[v_1] = 1|v_1\right] \cdot \mathbb{P}\left[B^*[v_2] = 0|v_1\right]}{\mathbb{P}\left[B^*[v_1] = 1|v_2\right] \cdot \mathbb{P}\left[B^*[v_2] = 0|v_2\right]} = \frac{p(1 - q)}{q(1 - p)} \leq e^\epsilon \tag{24}$$

Therefore, letting the equal sign in Equation (24) hold, we can set p as follows:

$$p = \frac{qe^{\epsilon}}{1 - q + qe^{\epsilon}}. \tag{25}$$

Applying Equation (25) to Equation (13), the variance of UE is

$$\mathrm{Var}_{UE}^{*}[\widehat{f_v}] = \frac{(1 - q + qe^{\epsilon})^2}{Nq(1 - q)(e^{\epsilon} - 1)^2}. \tag{26}$$

Symmetric UE (SUE) [35] is the symmetric version of UE when choosing p and q such that $p + q = 1$. Based on this observation and Equation (25), we can derive $p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$ and $q = \frac{1}{e^{\epsilon/2}+1}$. Then, the frequency can be computed based on Equation (21). The variance of SUE is

$$\mathrm{Var}_{SUE}^{*}[\widehat{f_v}] = \frac{e^{\epsilon/2}}{N(e^{\epsilon/2} - 1)^2}. \tag{27}$$

Optimized UE (OUE) [35] is to minimize the Equation (26). By making the partial derivative of Equation (26) with respect to q equals to 0, we can get the formula $\frac{1}{(e^{\epsilon}-1)^2}\left(\frac{e^{2\epsilon}}{(1-q)^2} - \frac{1}{q^2}\right)$. By solving this, we can obtain

$$p = \frac{1}{2}, \; q = \frac{1}{e^{\epsilon} + 1}. \tag{28}$$

The estimated frequency can be computed by Equation (21). By combining the Equations (26) and (28), the variance of OUE is

$$\mathrm{Var}_{OUE}^{*}[\widehat{f_v}] = \frac{4e^{\epsilon}}{N(e^{\epsilon} - 1)^2}. \tag{29}$$

3.1.3. Hash Encoding

In the same way of UE, Basic RAPPOR [27] encodes each value $v \in \mathcal{K}$ into a length-k binary bit vector B and conducts Randomization with the following two steps.

Step 1: Permanent randomized response. Generate B_1 with the probability

$$\mathbb{P}\left[B_1[v] = 1\right] = \begin{cases} 1 - \frac{1}{2}r, & \text{if } B[v] = 1, \\ \frac{1}{2}r, & \text{if } B[v] = 0. \end{cases} \tag{30}$$

where r is a user-tunable parameter that controls the level of longitudinal privacy guarantee.

Step 2: Instantaneous randomized response. Perturb B_1 with the following probability distribution (i.e., UE)

$$\mathbb{P}\left[B^*[i] = 1\right] = \begin{cases} p, & \text{if } B_1[i] = 1, \\ q, & \text{if } B_1[i] = 0. \end{cases} \tag{31}$$

From the proof in [27], the Permanent randomized response (i.e., *Step 1*) achieves ϵ-LDP for $\epsilon = 2\ln\frac{1-r/2}{r/2}$. The communication and computing cost of Basic RAPPOR is $\Theta(k)$ for each user, and $\Theta(Nk)$ for the aggregator. However, Basic RAPPOR does not scale to the cardinality k.

RAPPOR [27] adopts Bloom filters [66] to encode each single element based on a set of m hash functions $\mathbb{H} = \{H_1, H_2, \cdots, H_m\}$. Each hash function firstly outputs an integer in $\{0, 1, \cdots, k-1\}$. Then, each value v is encoded as a k-bit binary vector B by

$$B[i] = \begin{cases} 1, & \text{if } \exists H \in \mathbb{H}, s.t., H(v) = i, \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

Next, RAPPOR uses the same processes (i.e., Equations (30) and (31)) as Basic RAPPOR to conduct randomization.

From the proof in [27], RAPPOR achieves ϵ-LDP for $\epsilon = 2m \ln \frac{1-r/2}{r/2}$. Moreover, the communication cost of RAPPOR is $\Theta(k)$ for each user. However, the computation cost of the aggregator in RAPPOR is higher than Basic RAPPOR due to the LASSO regression.

O-RAPPOR [64] is proposed to address the problem of holding no prior knowledge about the attribute domain. Kairouz et al. [64] examined discrete distribution estimation when the open alphabets of categorical attributes are not enumerable in advance. They applied hash functions to map the underlying values at first. Then, the hashed values will be involved in a perturbation process, which is independent of the original values. On the basis of RAPPOR, Kairouz et al. adopted the idea of hash cohorts. Each user u_i will be assigned to a cohort c_i that is sampled i.i.d. from a uniform distribution over $\mathcal{C} = \{1, \cdots, C\}$. Each $c \in \mathcal{C}$ provides an independent view of the underlying distribution of strings. Based on hash cohorts, O-RAPPOR applies hash functions on a value v in cohort c before using RAPPOR and generates an independent h-bit hash Bloom filter $BLOOM_c$ for each cohort c, where the j-th bit of $BLOOM_c$ is 1 if $HASH_{c,h'}(v) = j$ for any $h' \in [1 \cdots h]$. Next, the perturbation on $BLOOM_c$ follows the same strategy in RAPPOR.

O-RR [64] is proposed to deal with non-binary attributes. It integrates hash cohorts into k-RR to deal with the case where the domain of attribute is unknown. Users in a cohort use their cohort hash function to project the value space into k disjoint subsets, i.e., $x_i = HASH_c(v) \mod k = HASH_c^k(v)$. Next, the O-RR perturbs the input value v as follows:

$$\mathbb{P}[v^*|v] = \frac{1}{C(e^\epsilon + k - 1)} \begin{cases} e^\epsilon, & \text{if } HASH_c^k(v) = v^*, \\ 1, & \text{if } HASH_c^k(v) \neq v^*. \end{cases} \tag{33}$$

Please note that Equation (33) contains a factor of C compared to Equation (17). This is because each value v belongs to one of the cohorts. The error bound of O-RR is the same as k-RR, but incurs more time cost due to hash and cohort operations.

To reduce communication and computation cost, local hashing (LH) [35] is proposed to hash the input value into a domain $[g]$ such that $g < k$. Denote $\mathbb{H}$ as the universal hash function family. Each input value is hashed into a value in $[g]$ by hash function $H \in \mathbb{H}$. The universal property requires that

$$\forall v_1, v_2 \in [k], v_1 \neq v_2 : \underset{H \in \mathbb{H}}{P}[H(v_1) = H(v_2)] \leq \frac{1}{g}. \tag{34}$$

Randomization. Given any input value $v \in [k]$, LH first outputs a value x in $[g]$ by hashing, i.e., $x = H(v)$. Then, LH perturbs x with the following distribution

$$\forall i \in [g], \mathbb{P}[y = i] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + g - 1}, & \text{if } x = i, \\ q = \frac{1}{e^\epsilon + g - 1}, & \text{if } x \neq i. \end{cases} \tag{35}$$

After perturbation, each user sends $\langle H, y \rangle$ to the aggregator. Based on Equation (35), we can know that LH satisfies ϵ-LDP since it always holds that $p \leq qe^\epsilon$.

Aggregation and Estimation. Assume we aim to estimate the frequency f_v of the value v. The aggregator counts the total number that $\langle H, y \rangle$ supports value v, denoted as θ. That is, for each report $\langle H, y \rangle$, if it holds that $H(v) = y$, then $\theta = \theta + 1$. Based on Equation (35), it holds that

$$p^* = p, \quad q^* = \frac{1}{g}p + \frac{g-1}{g}q = \frac{1}{g}, \tag{36}$$

where p^* is the probability of keeping unchanged of an input value and q^* is the probability of flipping an input value.

Then, while aggregating in the server, we have

$$f_v p^* + (1 - f_v)q^* = \theta/N. \tag{37}$$

Based on Equations (36) and (37), we can get the estimated frequency of the value v, i.e.,

$$\widehat{f_v} = \left(\frac{g\theta}{N} - 1 \right) \cdot \frac{e^\epsilon + g - 1}{g e^\epsilon - e^\epsilon - g + 1}. \tag{38}$$

By taking $p = p^*$, $q = q^*$ into Equation (13), the variance of LH is

$$\mathrm{Var}^*_{\mathrm{LH}}[\widehat{f_v}] = \frac{(e^\epsilon + g - 1)^2}{N(g-1)(e^\epsilon - 1)^2}. \tag{39}$$

Local hashing will become Binary Local Hashing (BLH) [35] when $g = 2$. In BLH, each hash function $H \in \mathbb{H}$ hashes an input from $[k]$ into one bit.

Randomization. Based on Equation (35), the randomization of BLH follows the probability distribution as

$$\mathbb{P}\left[y = 1\right] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon+1}, & \text{if } x = 1, \\ q = \frac{1}{e^\epsilon+1}, & \text{if } x = 0. \end{cases} \tag{40}$$

Aggregation and Estimation. Based on LH, it holds that $p^* = p$ and $q^* = \frac{1}{2}p + \frac{1}{2}q = \frac{1}{2}$. When the reported supports of value v is θ, based on Equation (38). the estimated frequency can be computed as

$$\widehat{f_v} = \left(\frac{2\theta}{N} - 1 \right) \cdot \frac{e^\epsilon + 1}{e^\epsilon - 1}. \tag{41}$$

The variance of BLH is

$$\mathrm{Var}^*_{\mathrm{BLH}}[\widehat{f_v}] = \frac{(e^\epsilon + 1)^2}{N(e^\epsilon - 1)^2}. \tag{42}$$

Optimized LH (OLH) [35] aims to choose an optimized g to compromise the information losses between hash step and randomization step. Based on Equation (39), we can minimize the variance of LH by making the partial derivative of Equation (39) with respect to g equals to 0. That is, it is equivalent to solve the following equation $(e^\epsilon - 1)^2 \cdot g - (e^\epsilon - 1)^2(e^\epsilon + 1) = 0$. By solving it, the optimal g is $g = e^\epsilon + 1$, where $g = \lfloor e^\epsilon + 1 \rfloor$ in practice. When the reported supports of value v is θ, based on Equation (38), the estimated frequency is $\widehat{f_v} = \frac{2(g\theta - N)}{N(e^\epsilon - 1)}$. In addition, the variance of OLH is

$$\mathrm{Var}^*_{\mathrm{OLH}}[\widehat{f_v}] = \frac{4e^\epsilon}{N(e^\epsilon - 1)^2}. \tag{43}$$

3.1.4. Transformation

The transformation-based method is usually adopted to reduce the communication cost.

S-Hist [61] is proposed to produce a succinct histogram that contains the most frequent items (i.e., "heavy hitters"). Bassily and Smith [61] have proved that S-Hist achieves asymptotically optimal accuracy for succinct histogram estimation. S-Hist randomly selects only one bit from the encoded vector based on random matrix projection, which reduces the communication cost. The specific process of S-Hist is as follows, which includes an additional initialization step.

Initialization. The aggregator generates a random projection matrix $\Phi \in \{-\frac{1}{\sqrt{b}}, \frac{1}{\sqrt{b}}\}^{b \times k}$, where each element of Φ is extracted from the set $\{-\frac{1}{\sqrt{b}}, \frac{1}{\sqrt{b}}\}$. The magnitude of each column vector in Φ is 1, and the inner product of any two different column vectors is 0. Here b is a constant parameter determined by error bound, where error is defined as the maximum distance between the estimated and true frequencies, i.e., $\max_{v \in \mathcal{K}} |\widehat{f}_v - f_v|$.

Randomization. Assume the input value v is the v-th element of domain $\mathcal{K}$. We encode v as $Encode(v) = \langle j, x \rangle$, where j is chosen uniformly at random from $[b]$, and x is the v-th element of the j-th row of Φ, i.e, $x = \Phi[j, v]$. Then, we randomize x as follows:

$$z = \begin{cases} c_\epsilon bx, & \text{w.p. } \frac{e^\epsilon}{e^\epsilon + 1}, \\ -c_\epsilon bx, & \text{w.p. } \frac{1}{e^\epsilon + 1}, \end{cases} \tag{44}$$

where $c_\epsilon = \frac{e^\epsilon + 1}{e^\epsilon - 1}$. After perturbation, each user $u_i (i \in [1, N])$ sends $\langle j_i, z_i \rangle$ to the aggregator.

Aggregation and Estimation. Upon receiving the report $\langle j^i, z^i \rangle$ of each user u_i, the estimation for the v-th element of $\mathcal{K}$ is computed by

$$\widehat{f}_v = \sum_{i \in [1, N]} z^i \cdot \Phi[j^i, v]. \tag{45}$$

Based on Equation (44), it is easy to know that S-Hist satisfies ϵ-LDP for every choice of the index j. Furthermore, Bassily and Smith [61] proved that the L_∞-error of S-Hist is bounded by $O\left(\frac{1}{\epsilon}\sqrt{\frac{\log(k/\beta)}{N}}\right)$ with probability as least $1 - \beta$.

Hadamard Randomized Response (HRR) [26,31,67] is a useful tool to handle sparsity by transforming the information contained in sparse vectors into a different orthonormal basis. HRR adopts Hadamard transformation (HT) to handle the situation where the inputs and marginals of individual users are sparse. HT is also called discrete Fourier transform, which is described by an orthogonal and symmetric matrix ϕ with dimension $2^k \times 2^k$. Each row/column in ϕ is denoted as $\phi_{i,j} = 2^{-k/2}(-1)^{\langle i,j \rangle}$, where $\langle i, j \rangle$ denotes the number of 1's that i and j agree on in their binary representation. When a value v_i is presented as a sparse binary vector B_i, the full Hadamard transformation of the input is the B_i-th column of ϕ, i.e., the Hadamard coefficient $o_i = \phi \times B_i$.

Randomization. User i samples an index $j \in 2^k$ and perturbs $\phi_{B_i, j} \in \{-1, 1\}$ by using BRR that keeps true value with probability p and flips the value with probability $1 - p$. Then, the user i reports the perturbed coefficient $\widehat{o}_i$ and the index j to the aggregator. As we can see, the communication cost is $O(\log k + 1) = O(\log k)$.

Aggregation and Estimation. Assume the observed sum of all received perturbed coefficient with index j is O_j. Then, the unbiased estimation of the j-th Hadamard coefficient $\widehat{o}_j$ (with the $2^{-k/2}$ factor scaled) is computed by

$$\widehat{o}_j = \frac{O_j}{2^{k/2}(2p - 1)}. \tag{46}$$

In this way, the aggregator can compute the unbiased estimations of all coefficients and apply inverse transformation to produce the final frequency estimation $\widehat{f}$.

Based on the proof in [31,68], the variance of HRR is

$$\mathrm{Var}_{\mathrm{HRR}}[\widehat{f}] = \frac{4p(1-p)}{N(2p-1)^2}. \tag{47}$$

By setting $p = \frac{e^\epsilon}{e^\epsilon+1}$ to ensure LDP, the variance is $\frac{4e^\epsilon}{N(e^\epsilon-1)^2}$. Thus, HRR provides a good compromise between accuracy and communication cost. Besides, the computation overhead in the aggregator is $O(N + k\log k)$, versus $O(Nk)$ for OLH.

Furthermore, Jayadev et al. [67] designed a general family of LDP schemes. Based on Hadamard matrices, they choose the optimal privatization scheme from the family for high privacy with less communication cost and higher efficiency.

3.1.5. Subset Selection

The main idea of subset selection is randomly select ω items from the domain $\mathcal{K}$.

ω-Subset Mechanism (ω-SM) [69,70] is proposed to randomly reports a subset z with size ω of the original attribute domain $\mathcal{K}$, i.e., $z \subseteq \mathcal{K}$. Essentially, the output space z is the power set of the data domain $\mathcal{K}$. In addition the conditional probabilities of any input $v \in \mathcal{K}$, output $z \subseteq \mathcal{K}$ are as follows:

$$\mathbb{P}[z|v] = \begin{cases} \frac{\omega e^\epsilon}{\omega e^\epsilon+k-\omega}/\binom{k}{\omega}, & \text{if } |z| = \omega \text{ and } v \in z, \\ \frac{\omega}{\omega e^\epsilon+k-\omega}/\binom{k}{\omega}, & \text{if } |z| = \omega \text{ and } v \notin z, \\ 0, & \text{if } |z| \neq \omega. \end{cases} \tag{48}$$

As we can see, when $\omega = 1$, the 1-SM is equivalent to generalized randomized response (GRR) mechanism.

Randomization. Based on Equation (48), the randomization procedure of ω-SM is shown in Algorithm 2. Observe that the core part of randomization is randomly sampling $\omega - 1$ or ω elements from $\mathcal{K} - \{v\}$ without replacement.

Algorithm 2: The Randomization of ω-SM

Input: Input $v \in \mathcal{K}$, privacy budget ϵ, subset size ω
Output: Subset $z \subseteq \mathcal{K}$ with size ω

1 Initialize z as an empty set, $z = \{\}$;
2 Sample a Bernoulli variable u with $\mathbb{P}[u = 1] = \frac{\omega e^\epsilon}{\omega e^\epsilon+k-\omega}$;
3 **if** $u = 1$ **then**
4   Insert v into z, $z = z \cup \{v\}$;
5   Randomly sample $\omega - 1$ elements Y from $\mathcal{K} - \{v\}$;
6   Add elements in Y to z, $z = z \cup Y$;
7 **else**
8   Randomly sample ω elements Y from $\mathcal{K} - \{v\}$;
9   Add elements in Y to z, $z = z \cup Y$;
10 **return** z;

Aggregation and Estimation. Denote f_i and $\bar{f}_i$ as the real and the received frequency of the i-th value v_i, respectively. Upon receiving a private view z_i, it will increase $\bar{f}_i$ for each $v_i \in z_i$. Based on Algorithm 2, we can know that the true positive rate θ_{tr} is $\theta_{tr} = \frac{\omega e^\epsilon}{\omega e^\epsilon+k-\omega}$, which is the probability of v_i staying unchanged when the input value is v_i. In addition the false positive rate θ_{fr} is $\theta_{fr} = \frac{\omega e^\epsilon}{\omega e^\epsilon+k-\omega} \cdot \frac{\omega-1}{k-1} + \frac{k-\omega}{\omega e^\epsilon+k-\omega} \cdot \frac{\omega}{k-1}$, which is the probability of $v_{i'}$ showing in the private view when the input value is v_i $(i \neq i')$. Therefore, the expectation of $\bar{f}_i$ is $\mathbb{E}[\bar{f}_i] = f_i \cdot \theta_{tr} + (1 - f_i) \cdot \theta_{fr}$.

Thus, we can get the estimated frequency of $\widehat{f}_i$ is

$$\widehat{f}_i = \frac{\bar{f}_i - \theta_{fr}}{\theta_{tr} - \theta_{fr}}. \tag{49}$$

Comparisons. Table 4 summarizes the general LDP protocols from the perspective of encoding principles. The error bound is measured by L_∞-norm. BRR and GRR are direct perturbation-based methods, which are suitable for low-dimensional data. BRR has a communication cost of $O(\log 2) = O(1)$ and has a smaller error bound than other mechanisms. GRR is the general version of BRR when $k > 2$, of which the communication cost and error bound are both sensitive to domain size k. Both SUE and OUE are unary encoding-based methods. They have the same communication cost and error bound. RAPPOR, O-RAPPOR, O-RR, BLH, and OLH are hash encoding-based methods. RAPPOR, O-RAPPOR, and O-RR have larger error bounds and are relatively harder to implement since they involve Bloom filters and hash cohorts. BLH and OLH have smaller error bounds and are applicable to all privacy regime and any attribute domain. S-Hist and HRR are transformation-based methods, which have a lower communication cost. ω-SM is a subset selection-based method, which reduces the communication cost.

Table 4. Comparisons of general LDP protocols for frequency estimation.

Encoding Principle	LDP Algo.	Comm. Cost	Error Bound	Variance	Know Domain?
Direct Perturbation	BRR [43,63]	$O(1)$	$O(\frac{1}{\epsilon\sqrt{N}})$	$\frac{e^\epsilon}{N(e^\epsilon-1)^2}$	Y
	GRR [64] (or, DE/k-RR)	$O(\log k)$	$O(\frac{\sqrt{k\log k}}{\epsilon\sqrt{N}})$	$\frac{e^\epsilon+k-2}{N(e^\epsilon-1)^2}$	Y
Unary Encoding	SUE [35]	$O(k)$	$O(\frac{\sqrt{\log k}}{\epsilon\sqrt{N}})$	$\frac{e^{\epsilon/2}}{N(e^{\epsilon/2}-1)^2}$	Y
	OUE [35]	$O(k)$	$O(\frac{\sqrt{\log k}}{\epsilon\sqrt{N}})$	$\frac{4e^\epsilon}{N(e^\epsilon-1)^2}$	Y
Hash Encoding	RAPPOR [27]	$\Theta(k)$	$O(\frac{k}{\epsilon\sqrt{N}})$	$\frac{e^{\epsilon/2}}{N(e^{\epsilon/2}-1)^2}$	Y
	O-RAPPOR [64]	$\Theta(k)$	$O(\frac{k}{\epsilon\sqrt{N}})$	$\frac{e^{\epsilon/2}}{N(e^{\epsilon/2}-1)^2}$	N
	O-RR [64]	$O(\log k)$	$O(\frac{\sqrt{k\log k}}{\epsilon\sqrt{N}})$	$\frac{e^\epsilon+k-2}{N(e^\epsilon-1)^2}$	N
	BLH [35]	$O(\log k)$	$O(\frac{\sqrt{\log k}}{\epsilon\sqrt{N}})$	$\frac{(e^\epsilon+1)^2}{N(e^\epsilon-1)^2}$	Y
	OLH [35]	$O(\log k)$	$O(\frac{\sqrt{\log k}}{\epsilon\sqrt{N}})$	$\frac{4e^\epsilon}{N(e^\epsilon-1)^2}$	Y
Transformation	S-Hist [61]	$O(\log b)$	$O(\frac{\sqrt{\log k}}{\epsilon\sqrt{N}})$	$\frac{e^\epsilon}{N(e^\epsilon-1)^2}$	Y
	HRR [31]	$O(\log k)$	$O(\frac{\sqrt{\log k}}{\epsilon\sqrt{N}})$	$\frac{4e^\epsilon}{N(e^\epsilon-1)^2}$	Y
Subset Selection	ω-SM [69,70]	$O(\omega)$	$O(\frac{\sqrt{k\log k}}{\epsilon\sqrt{N}})$	$\frac{e^\epsilon+k-2}{N(e^\epsilon-1)^2}$	Y

Discussions. Section 3.1 discusses the general frequency estimation protocols with LDP. When focusing on multiple attributes (i.e., d-dimensional data), we can directly use the above protocols to estimate the frequency of each attribute. We can also estimate joint frequency distributions of multiple attributes by using the above protocols as long as we regard the Cartesian product of the values of multiple attributes as the total domain. However, the error bound will be d times greater than dealing with a single attribute. Even worse is that the total domain cardinality will increase exponentially with the dimension d, which leads to huge computation overhead and low data utility. Therefore, several studies [31,71–75] have investigated on estimating joint probability distributions of d-dimensional data, which are summarized in Section 3.6.

3.2. Frequency Estimation on Set-Valued Data

This section summarizes the mechanisms for frequency estimation on set-valued data, including items distribution estimation, frequent items mining, and frequent itemsets mining.

Set-valued data denotes a set of items. Let $\mathcal{K} = \{v_1, v_2, \cdots, v_k\}$ be the domain of items. Set-valued data V^i of user i is denoted as a subset of $\mathcal{K}$, i.e., $V^i \subseteq \mathcal{K}$. Different users may have different number of items. Table 5 shows a set-valued dataset of six users with item domain $\mathcal{K} = \{A, B, C, D, E, F\}$. In what follows, we will introduce each frequency-based task on set-valued data.

Table 5. Example of set-valued dataset.

V^1	$\{A, C, E\}$
V^2	$\{B, D, E\}$
V^3	$\{A, B, E\}$
V^4	$\{A, D, E\}$
V^5	$\{A, D, F\}$
V^6	$\{A, F\}$

3.2.1. Item Distribution Estimation

The basic frequency estimation task on set-valued data is to analyze the distributions over k items. For an item $v \in \mathcal{K}$, its frequency is defined as the fraction of times that v occurs. That is, $f_v := \frac{1}{N} |\{V^i | v \in V^i\}|$. Let c_v be the number of users whose data include v. Then, we have $c_v := |\{V^i | v \in V^i\}|$ and $f_v = \frac{1}{N} c_v$.

To tackle set-valued data with LDP, there are two tough challenges [76]. (*i*) *Huge domain*: supposing there are total k items and each user has at most l items, then the number of possible combinations of items for each user could reach to $\binom{k}{l} + \binom{k}{l-1} + \cdots + \binom{k}{0}$. (*ii*) *Heterogeneous size*: different users may have different numbers of items, varying from 0 to l. To address heterogeneous size, one of the most general methods is to add the padding items to the below-sized record. Then, the naive method is to treat the set-valued data as categorical data and use LDP algorithms in Section 3.1. However, this method needs to divide the privacy budget into smaller parts, thereby introducing excessive noise and reducing data utility.

Wang et al. [76] proposed PrivSet mechanism which has a linear computational overhead with respect to the item domain size. PrivSet pre-processed the number of items of each user to l, which addresses the issue of heterogeneous size. When the item size is beyond l, it simply truncates or randomly samples l items from the original set. When the item size is under l, it adds padding items to the original set. After pre-processing, each padded set-valued data $\widehat{s}$ belongs to $\widehat{S} = \{b | b \subseteq \widehat{\mathcal{K}} \text{ and } |b| = l\}$, where $\widehat{\mathcal{K}}$ is the item domain after padding. Then, PrivSet randomized data based on the exponential mechanism. For each padded data $\widehat{s} \in \widehat{S}$, the randomization component of PrivSet selects and outputs an element $\widehat{t} \in \widehat{T} = \{a | a \subseteq \widehat{\mathcal{K}} \text{ and } |a| = l'\}$ with probability

$$p = \begin{cases} \frac{exp(\epsilon)}{\Omega}, & \text{if } \widehat{s} \cap \widehat{t} \neq \varnothing, \\ \frac{1}{\Omega}, & \text{if } \widehat{s} \cap \widehat{t} = \varnothing, \end{cases} \tag{50}$$

where Ω is the probability normalizer and equals to $\binom{k}{l'} + exp(\epsilon) \cdot \left(\binom{k+l}{l'} - \binom{k}{l'}\right)$. As analyzed in [76], the randomization of PrivSet reduces the computation cost from $O\left(\binom{k+l}{l}\right)$ to $O(k)$, which is linear to item domain size. PrivSet also holds a lower error bound over other mechanisms.

Moreover, LDPart [77] is proposed to generate sanitized location-record data with LDP, where location-record data are treated as a special case of set-valued data. LDPart uses a partition tree to greatly reduce the domain space and leverages OUE [35] to perturb the input values. However,

the utility of LDPart quite relies on two parameters, i.e., the counting threshold and the maximum length of the record. It's very difficult to calculate the optimal values of these two parameters.

3.2.2. Frequent Items Mining

Frequent items mining (also known as heavy hitters identification, top-ω hitters mining, or frequent terms discovery) has played important roles in data statistics. Based on the notations in Section 3.2.1, we say an item v is ω-heavy (ω-frequent) if its multiplicity is at least ω, i.e., $c_v \geq \omega$. The task of frequent items mining is to identify all ω-heavy hitters from the collected data. For example, as shown in Table 5, the 3-heavy items are A, D, and E.

Bassily and Smith [61] focused on producing a succinct histogram that contains the most frequent items of the data under LDP. They leveraged the random matrix projection to achieve much lower communication cost and error bound than that of earlier methods in [78,79]. The specific process of S-Hist is introduced in Section 3.1. Furthermore, a follow-up work in [80] proposed TreeHist that computes heavy hitters from a large domain. TreeHist transforms the user's value into a binary string and constructs a binary prefix tree to compute frequent strings, which improves both efficiency and accuracy. Moreover, with strong theoretical analysis, Bun et al. [81] proposed a new heavy hitter mining algorithm that achieves the optimal worst-case error as a function of the domain size, the user number, the privacy budget, and the failure probability.

To address the challenge that the number of items in each user record is different, Qin et al. [30] proposed a Padding-and-Sampling frequency oracle (PSFO) that first pads user's items into a uniform length l by adding some dummy items and then makes each user sample one item from possessed items with the same sampling rate. They designed LDPMiner based on RAPPOR [27] and S-Hist [61]. LDPMiner adopts a two-phase strategy with privacy budgets ϵ_1 and ϵ_2, respectively. In phase 1, LDPMiner identifies the potential candidate set of frequent items (i.e., the top-ω frequent items) by using a randomized protocol with ϵ_1. The aggregator broadcasts the candidate set to all users. In phase 2, LDPMiner refines the frequent items from the candidates with the remaining privacy budget ϵ_2 and outputs the frequencies of the final frequent items. LDPMiner is much wiser on budget allocation than naive method. However, LDPMiner still needs to split the privacy budget into $2l$ parts at both phases, which limits the data utility.

Wang et al. [75] proposed a prefix extending method (PEM) to discover heavy hitters from an extremely large domains (e.g., $k = 2^{128}$). To address the computing challenge, PEM iteratively identifies the increasingly longer frequent prefixes based on a binary prefix tree. Specifically, PEM first divides users into g equal-size groups, making each group i ($1 \leq i \leq g$) associate with a particular prefix length s_i, such that $1 < s_1 < s_i < \cdots < s_g = \lceil \log k \rceil$. Then each user reports the private value using LDP protocol and the server iterates through the groups. Obviously, group size g is a key parameter that influences both the computation complexity and data utility. Therefore, Wang et al. [75] further designed a sensitivity threshold principle that computes a threshold to control the false positives, thus maintaining the effectiveness and accuracy of PEM. Jia et al. [82] have pointed out that prior knowledge can be used to improve the data utility of the LDP algorithms. Thus, `Calibrate` was designed to incorporate the prior knowledge via statistical inference, which can be appended to the existing LDP algorithms to reduce estimation errors and improve the data utility.

3.2.3. Frequent Itemset Mining

Frequent itemset mining is much similar to frequent items mining, except that the desired results of the former will become a set of itemsets rather than items. Frequent itemsets mining is much more challenging since the domain size of itemsets is exponentially increased.

Following the definition of frequent item mining in Section 3.2.2, the frequency of any itemset $\mathbf{v} \subseteq \mathcal{K}$ is defined as the fraction of times that itemset $\mathbf{v}$ occurs. That is, $f_{\mathbf{v}} := \frac{1}{N}|\{V^i|\mathbf{v} \subseteq V^i\}|$. The count of any itemset $\mathbf{v} \subseteq \mathcal{K}$ is defined as the total number of users whose data include $\mathbf{v}$ as a subset. That is, $c_{\mathbf{v}} := |\{V^i|\mathbf{v} \subseteq V^i\}|$. An ω-heavy itemset $\mathbf{v}$ is such that its multiplicity is at least ω,

i.e., $c_{\mathbf{v}} \geq \omega$. The task of frequent itemset mining is to identify all ω-heavy itemsets from the collected data. For example, the 3-heavy itemsets in Table 5 are $\{A\}$, $\{D\}$, $\{E\}$, and $\{A, E\}$.

Sun et al. [83] proposed a personalized frequent itemset mining algorithm that provides different privacy levels for different items under LDP. They leveraged the randomized response technique [43] to distort the original data with personalized privacy parameters and reconstructed itemset supports from the distorted data. This method distorted each item in domain separately, which leads to an error bound of $O(\frac{k\sqrt{\log k}}{\epsilon\sqrt{N}})$ that is super-linear to k. As introduced in Section 3.2.2, LDPMiner mines frequent items over set-valued data by using a PSFO protocol. Inspired by LDPMiner, Wang et al. [84] also padded users items into size l and sampled one item from the possessed items of each user, which could ensure the frequent items can be reported with high probability even though there still exist unsampled items. Specifically, Wang et al. [84] designed a Set-Value Item Mining (SVIM) protocol, and then based on the results from SVIM, they proposed a more efficient Set-Value ItemSet Mining (SVSM) protocol to find frequent itemsets. To further improve the data utility of SVSM, they also investigated the best-performing-based LDP protocol for each usage of PSFO by identifying the privacy amplification property of each LDP protocol.

3.2.4. New Terms Discovery

This section introduces the task of new terms discovery that focuses on the situation where the global knowledge of item domain is unknown. Discovering top-ω frequent new terms is an important problem for updating a user-friendly mobile operating system by suggesting words based on a dictionary.

Apple iOS [26,85], macOS [86], and Google Chrome [27,87] have integrated with LDP to protect users privacy when collecting and analyzing data. RAPPOR [27] is first used for frequency estimation under LDP, which is introduced in Section 3.1. Afterward, its augmented version A-RAPPOR [87] is proposed and applied in the Google Chrome browser for discovering frequent new terms. A-RAPPOR reduces the huge domain of possible new terms by collecting n-grams instead of full terms. Suppose the character domain is C. Then there will be C/n such n-gram groups. For each group, A-RAPPOR constructs the significant n-grams that will be used to construct a m-partite graph, where $m = C/n$. Thus, the frequent new terms can be found efficiently by finding m-cliques in a m-partite graph. However, A-RAPPOR has rather low utility since the n-grams cannot always represent the real terms. The variance of each group is limited as $\frac{4e^{\epsilon}}{N(e^{\epsilon}-1)^2}$.

To improve the accuracy and reduce the huge computational cost, Wang et al. [88] proposed PrivTrie which leverages an LDP-complaint algorithm to iteratively construct a trie. When constructing a trie, the naive method is to uniformly allocate the privacy budget to each level of the trie. However, this will lead to inaccurate frequency estimations, especially when the height of the trie is large. To address this challenge, PrivTrie only requires to estimate a coarse-grained frequency for each prefix based on an adaptive user grouping strategy, thereby remaining more privacy budget for actual terms. PrivTrie further enforces consistency in the estimated values to refine the noisy estimations. Therefore, PrivTrie achieved much higher accuracy and outperformed the previous mechanisms. Besides, Kim et al. [89] proposed a novel algorithm called CCE (Circular Chain Encoding) to discover new words from keystroke data under LDP. CCE leveraged the chain rule of n-grams and a fingerprint-based filtering process to improve computational efficiency.

Comparisons and discussions. Table 6 shows the comparisons of LDP-based protocols for frequency estimation on set-valued data, including communication cost, key technique, and whether need to know the domain in advance. As for set-valued data, the padding-and-sampling technique is always adopted to solve the problem of the heterogeneous item size of different users, such as in [30,76,84]. The padding size l is a key parameter for both efficiency and accuracy. How to choose an optimal l needs further study. Besides, tree-based method is also widely used in[75,77,80,88] to reconstruct set-valued data. The tree-based method usually requires partition users into different groups or partition privacy budget for each level of the tree, which limits the data utility. In this case,

the optimal budget allocation strategy needs to be designed. Meanwhile, the adaptive user grouping technique is also a better way to improve data utility, such as in [88].

Table 6. Comparisons of frequency estimation mechanisms for set-valued data with LDP.

Task	LDP Algorithm	Comm. Cost	Key Technique	Know Domain?
Item distribution estimation	PrivSet [76]	$O(l')$ [1]	Padding-and-sampling; Subset selection	Y
	LDPart [77]	$O(|V|_m)$ [2]	Tree-based (partition tree); Users grouping	Y
Frequent items mining	TreeHist [80]	$O(1)$	Tree-based (binary prefix tree)	Y
	LDPMiner [30]	$O(\log k + \omega)$	Padding-and-sampling; Wiser budget allocation	Y
	PEM [75]	$O(\log k)$	Tree-based (binary prefix tree); Users grouping	Y
	Calibrate [82]	$O(k)$	Consider prior knowledge	Y
Frequent itemset mining	Personalized [83]	$O(k)$	Personalized privacy regime	Y
	SVSM [84]	$O(\log k)$	Padding-and-sampling; Privacy amplification	Y
New terms discovering	A-RAPPOR [87]	$O(\log k)$	Select n-grams; Construct partite graph	N
	PrivTrie [88]	$O(|V|_m)$	Tree-based (trie); Adaptive users grouping; Consistency constraints	N

[1] l' is the output size of randomization, which is smaller than total domain size k. [2] $|V|_m$ is the maximum number of nodes among all layers of the tree.

3.3. Frequency Estimation on Key-Value Data

Key-value data [90] is such data that has a key-value pair including a key and a value, which is commonly used in big data analysis. For example, the key-value pairs (KV pair) that denote diseases and their diagnosis values are listed as $\{\langle Cancer, 0.3\rangle\}$, $\{\langle Fever, 0.06\rangle\}$, $\{\langle Flu, 0.6\rangle\}$, etc. While collecting and analyzing key-value data, there are four challenges to consider. (i) Key-value data contain two heterogeneous dimensions. The existing studies mostly focus on homogeneous data. (ii) There are inherent correlations between keys and values. The naive method that deals with key-value data by separately estimating the frequency of key and the mean of value under LDP will lead to the poor utility. (iii) One user may possess multiple key-value pairs that need to consume more privacy budget, resulting in larger noise. (iv) The overall correlated perturbation mechanism on key-value data should consume less privacy budget than two independent perturbation mechanisms for key and value respectively. We can improve data utility by computing the actually consumed privacy budget.

Ye et al. [90] proposed *PrivKV* that retains the correlations between keys and values while achieving LDP. *PrivKV* adopts *Harmony*[91] to perturb the value v of a KV pair into v^* with privacy budget ϵ_2 and converts the pair $\langle key, v^*\rangle$ into canonical form $\langle 1, v^*\rangle$ that is perturbed by

$$\langle key, v^*\rangle = \begin{cases} \langle 1, v^*\rangle, & \text{w.p. } \frac{e^{\epsilon_1}}{e^{\epsilon_1}+1}, \\ \langle 0, 0\rangle, & \text{w.p. } \frac{1}{e^{\epsilon_1}+1}. \end{cases} \tag{51}$$

Noted that in *PrivKV*, the value of a key-value pair is randomly drawn from the domain of $[-1, 1]$ when users do not own a key-value pair (i.e., the users have no apriori knowledge about the distribution of true values). Thus, *PrivKV* suffers from low accuracy and instability. Therefore, Ye et al. [90] built two algorithms *PrivKVM* and *PrivKVM$^+$* by multiple iterations to address this problem. Intuitively, as the number of iterations increases, the accuracy will be improved since the distribution of the values will be close to the distributions of the true values.

Based on *PrivKV*, Sun et al. [92] proposed several mechanisms for key-value data collection based on direct encoding and unary encoding techniques [35]. Sun et al. [92] also introduced conditional analysis for key-value data for the first time. Specifically, they proposed several mechanisms that support L-way conditional frequency and mean estimation while ensuring good accuracy.

However, the studies in [90,92] lack exact considerations on challenges (iii) and (iv) mentioned previously. On the one hand, they simply sample a pair when facing multiple key-value data pairs, which cannot make full use of the whole data pairs and may not work well for a large domain. On the other hand, they neglect the privacy budget composition when considering the inherent correlations of key-value data, thus leading to limited data utility.

Gu et al. [93] proposed the correlated key/value perturbation mechanism which reduces the privacy budget consumption and enhances data utility. They designed a Padding-and-Sampling protocol for key-value data to deal with multiple pairs of each user. Thus, it is no longer necessary to sample a pair from the whole domain (e.g., *PrivKVM*[90]), but sample from the key-value pairs possessed by users, thus eliminating the affects of large domain size. Then, they proposed two protocols PCKV-UE by adopting unary encoding and PCKV-GRR by adopting the generalized randomized response. Rather than sequential composition, both PCKV-UE and PCKV-GRR involve a near-optimal correlated budget composition strategy, thereby minimizing the combined mean square error.

Comparisons and discussions. Table 7 shows the comparisons of frequency/mean estimations on key-value data with LDP. To solve the challenge of each user has multiple data pairs, simple sampling [90,92] is adopted to sample a pair from the whole domain and padding-and sampling [93] is adopted to sample a pair from the key-value pairs possessed by users. Besides, *PrivKVM* [90] and PCKV-UE/PCKV-GRR [93] consider the correlations between key and value by iteration and correlated perturbation, respectively. However, CondiFre [92] lacks of the consideration on learning correlations when conducting conditional analysis. Furthermore, both *PrivKV* [90] and CondiFre [92] achieve LDP based on two independent perturbations with fixed privacy budget by sequential composition. PCKV-UE/PCKV-GRR [93] holds a tighter privacy budget composition strategy that makes the optimal allocation of privacy budget.

Table 7. Comparisons of LDP-based protocols for frequency/mean estimation on key-value data.

LDP Algorithm	Goal	Address Multiple Pairs	Learn Correlations	Composition	Allocation of ϵ
PrivKVM [90]	Mean value of values; frequency of keys	Simple sampling	Mechanism iteration	Sequential	Fixed
CondiFre [92]	Mean value of values; frequency of keys; *L*-way conditional analysis	Simple sampling	Not consider	Sequential	Fixed
PCKV-UE/ PCKV-GRR [93]	Mean value of values; frequency of keys	Padding-and-sampling	Correlated perturbation	Tighter bound	Optimal

3.4. Frequency Estimation on Ordinal Data

Compared to categorical data, ordinal data has a linear ordering among categories, which is concluded as ordered categorical data, discrete numerical data (e.g., discrete sensor/metering data), and preference ranking data makes the optimal allocation of privacy budget.

When quantifying the indistinguishability of two ordinal data with LDP, the work in [53] measured the distance of two ordinal data by ϵ-geo-indistinguishability. That is, a mechanism satisfies ϵ-DP if it holds $\mathbb{P}[\mathcal{M}(X_i) \in \mathcal{Y}] \leq e^{\epsilon \cdot d(X_i, X_j)} \cdot \mathbb{P}[\mathcal{M}(X_j) \in \mathcal{Y}]$ for any possible pairs $X_i, X_j \in \mathcal{X}$. The distance $d(X_i, X_j)$ of ordinal data X_i and X_j can be measured by Manhattan distance or squared Euclidean distance. Based on ϵ-geo-indistinguishability, Wang et al. [70,94] proposed subset exponential mechanism (SEM) that is realized by a tweaked version of exponential mechanism [22]. Besides, they also proposed a circling subset exponential mechanism (CSEM) for ordinal data with uniform topology. For both SEM and CSEM, the authors have provided the theoretical error bounds to show their mechanisms can reduce nearly a fraction of $\exp(-\frac{\epsilon}{2})$ error for frequency estimation.

Preference ranking data is also one of the most common representations of personal data and highly sensitive in some applications, such as preference rankings on political or service quality. Essentially, preference ranking data can be regarded as categorical data that hold an order among

different items. Given an item set $\mathcal{K} = \{v_1, v_2, \cdots, v_k\}$, a preference ranking of $\mathcal{K}$ is an ordered list that contains all k items in $\mathcal{K}$. Denote a preference ranking as $\sigma = \langle \sigma(1), \sigma(2), \cdots, \sigma(k), \rangle$, where $\sigma(j) = v_i$ means that item v_i's rank under σ is j. The goal of collecting preference rankings is to estimate the distribution of all different rankings from N users. It can easily verify that the domain of all rankings is $k!$, which leads to excessive noises and low accuracy when k is large. Yang et al. [95] proposed SAFARI that approximates the overall distributions over a smaller domain that is chosen based on a riffle independent model. SAFARI greatly reduces the noise amount and improves data utility.

Voting data is to some extent a kind of preference ranking. By aggregating the preference rankings based on one of the certain positional voting rules (e.g., Borda, Nauru, Plurality [96–98]), we can obtain the collective decision makings. To avoid leaking personal preferences in a voting system, the work in [99] collected and aggregated voting data with LDP while ensuring the usefulness and soundness. Specifically, weighted sampling mechanism and additive mechanism are proposed for LDP-based voting aggregation under general positional voting rules. Compared to the naïve Laplace mechanism, weighted sampling mechanism and additive mechanism can reduce the maximum magnitude risk bound from $+\infty$ to $O(\frac{k^3}{N\epsilon})$ and $O(\frac{k^2}{N\epsilon})$, respectively, where k is the size of vote candidates (i.e., the domain size), N is the number of users.

As one of the fundamental data analysis primitives, range query aims at estimating the fractions or quantiles of the data within a specified interval [100,101], which is also an analysis task on ordinal data. The studies in [68,102] have proposed some approaches to support range queries with LDP while ensuring good accuracy. They designed two methods to describe and analyze the range queries based on hierarchical histograms and the Haar wavelet transform, respectively. Both methods use OLH [35] to achieve LDP with low communication cost and high accuracy. Besides, local d-privacy is a generalized notion of LDP under distance metric, which is adopted to assign different perturbation probabilities for different inputs based on the distance metrics. Gu et al. [103] used local d-privacy to support both range queries and frequency estimation. They proposed an optimization framework by solving the linear equations of perturbation probabilities rather than solving an optimization problem directly, which not only reduces computation cost but also makes the optimization problem always solvable when using d-privacy.

3.5. Frequency Estimation on Numeric Data

Most existing studies compute frequency estimations on categorical data. However, there are many numerical attributes in nature, such as income, age. Computing frequency estimation of numeric data also plays important role in reality.

For numerical distribution estimation with LDP, the naïve method is to discretize the numerical domain and apply the general LDP protocols directly. However, the data utility of the naïve method relies heavily on the granularity of discretization. Even worse is that an optimal discretization strategy depends on privacy parameters and the original distributions of the numeric attributes. Thus, it is a big challenge to find the optimal discretization strategy. Li et al. [104] used the ordered nature of the numerical domain to compromise a better trade-off between privacy and utility. They proposed a novel mechanism based on expectation-maximization and smoothing techniques, which improves the data utility significantly.

3.6. Marginal Release on Multi-Dimensional Data

The marginal table is "the workhorse of data analysis" [31]. When obtaining marginal tables of a set of attributes, we can learn the underlying distributions of multiple attributes, identify the correlated attributes, describe the probabilistic relationships between cause and effects. Thus, k-way marginal release was widely investigated with LDP [31,72,73].

Denote $A = \{A_1, A_2, \cdots, A_d\}$ as the d attributes of d-dimensional data. For each attribute A_j ($j = 1, 2, \cdots, d$), the domain of A_j is denoted as $\Omega_j = \{\omega_j^1, \omega_j^2, \cdots, \omega_j^{|\Omega_j|}\}$, where ω_j^i is the i-th value of Ω_j and $|\Omega_j|$ is the cardinality of Ω_j. The marginal table is defined as follows.

Definition 11 (Marginal Table [31]). *Given d-dimensional data, marginal operator $\mathcal{C}^\beta$ computes all frequencies of different attribute combinations that are decided by $\beta \in \{0,1\}^d$, where $|\beta|$ denotes the number of $1's$ in β, and $|\beta| = k \leq d$. The marginal table contains all the returned results of $\mathcal{C}^\beta$.*

Example 1. *When $d = 4$ and $\beta = 0110$, it means that we estimate the probability distributions of all combination of the second and third attributes. $\mathcal{C}^{0110}$ returns the frequency distributions of all combinations.*

The *k*-way marginal is the probability distributions of any *k* attributes in *d* attributes.

Definition 12 (*k*-way Marginal [31]). *The k-way marginal is the probability distributions of k attributes in d attributes, i.e., $|\beta| = k$. For a fixed k, the set of all possible k-way marginals correspond to all $\binom{d}{k}$ distinct ways of picking k attributes from d, which called full k-way marginals.*

The *k*-way marginal release is to estimate *k*-way marginal probability distribution for any *k* attributes $A_{j_1}, A_{j_2}, \cdots, A_{j_k}$ chosen from *A*. The *k*-way marginal distribution of attributes $A_{j_1}, A_{j_2}, \cdots, A_{j_k}$ is denoted as $P(A_{j_1} A_{j_2} \cdots A_{j_k})$. It has

$$P(A_{j_1} A_{j_2} \cdots A_{j_k}) \triangleq P(\omega_{j_1} \omega_{j_2} \cdots \omega_{j_k}) \tag{52}$$
$$\text{for } \forall \omega_{j_1} \in \Omega_{j_1}, \omega_{j_2} \in \Omega_{j_2}, \cdots, \omega_{j_k} \in \Omega_{j_k}.$$

3.6.1. *k*-Way Marginal Probability Distribution Estimation

Randomized response technique [27] can be naïvely leveraged to achieve LDP when computing *k*-way marginal probability distributions. However, both the efficiency and accuracy will be seriously affected by the "curse of high-dimensionality". The total domain cardinality will be $\prod_{j=1}^{k} |\Omega_j|$, which increases exponentially as *k* increases.

The EM-based algorithm with LDP [87] is restricted to 2-way marginals. When *k* is large, it will lead to high time/space overheads. The work in [71] proposed a Lasso-based regression mechanism that can estimate high-dimensional marginals efficiently by extracting key features with high probabilities. Besides, Ren et al. [72] proposed LoPub to find compactly correlated attributes to achieve dimensionality reduction, which further reduces the time overhead and improves data utility.

Nonetheless, the *k*-way marginals release still suffers from low data utility and high computational overhead when *k* becomes larger. To solve this, the work in [73] proposed to leverage Copula theory to synthesize multi-dimensional data with respect to marginal distributions and attribute dependence structure. It only needs to estimate one and two-marginal distributions instead of *k*-way marginals, thus circumventing the exponential growth of domain cardinality and avoiding the curse of dimensionality. Afterward, Wang et al. [105] further leveraged C-vine Copula to take the conditional dependencies among high-dimensional attributes into account, which significantly improves data utility.

Cormode et al. [31] have investigated marginal release under under different kinds of LDP protocols. They further proposed to materialize marginals by a collection of coefficients based on the Hadamard transform (HT) technique. The underlying motivation of using HT is that the computation of *k*-way marginals require only a few coefficients in the Fourier domain. Thus, this method improves the accuracy and reduces the communication cost. Nonetheless, this method is designed for binary attributes. The non-binary attributes need to be pre-processed to binary types, leading to higher dimensions. To further improve the accuracy, Zhang et al. [74] proposed a consistent adaptive local marginal (CALM) algorithm. CALM is inspired by PriView [106] that builds *k*-way marginal by taking the form of *m* marginals each of the size *l* (i.e., synopsis). Besides, the work in [107,108] focused on answering multi-dimensional analytical queries that are essentially formalized as computing *k*-way marginals with LDP.

Comparisons and discussions. Table 8 summarized the LDP-based algorithms for *k*-way marginal release. To improve efficiency, the existing methods try to reduce the large domain space by various techniques, such as HT and dimensionality reduction. As we can see, the variances of the existing methods are relatively large, leading to limited data utility. Although the subset selection is a useful way to reduce the communication cost and variance, it suffers from the sampling error when constructing low-dimensional synopsis. Therefore, designing mechanisms with high data utility and low costs still faces big challenges when *d* is large.

Table 8. Comparisons of LDP-based algorithms for *k*-way marginal release of *d*-dimensional data.

LDP Algorithm	Key Technique	Comm. Cost	Variance	Time Complexity	Space Complexity
RAPPOR [27]	Equal to naïve method when *d*>2	$O\left(\prod_{j=1}^{d} \lvert \Omega_j \rvert\right)$	$2^d \cdot \mathrm{Var}$	High	High
Fanti et al. [87]	Expectation Maximization (EM)	$O\left(\sum_{j=1}^{d} \lvert \Omega_j \rvert\right)$	$2^d \cdot \mathrm{Var}$	$O\left(N \cdot \sum_{i=1}^{k} \binom{d}{i} 2^i\right)$	$O\left(\sum_{i=1}^{k} \binom{d}{i} 2^i\right)$
LoPub [72]	Lasso regression; Dimensionality and sparsity reduction	$O\left(\sum_{j=1}^{d} \lvert \Omega_j \rvert\right)$	$2^d \cdot \mathrm{Var}$	Medium	High
Cormode et al. [31]	Hadamard Transformation (HT)	$O\left(\sum_{i=1}^{k} \binom{d}{i}\right)$	$\sum_{i=1}^{k} \binom{d}{i} \cdot \mathrm{Var}$	$O\left(N + \binom{d}{k} \cdot 2^k\right)$	$O\left(\sum_{i=1}^{k} \binom{d}{i}\right)$
LoCop [105]	Lasso-based regression; Attribute correlations learning	$O\left(\sum_{j=1}^{d} \lvert \Omega_j \rvert\right)$	$2^d \cdot \mathrm{Var}$	Low	High
CALM [74]	Subset selection; Consistency constraints	$O(2^l)$	$\frac{m}{N} \cdot 2^l \cdot \mathrm{Var}$	$O(N \cdot 2^l)$	$O(m \cdot 2^l)$

[1] Var is the variance of estimating a single cell in the full contingency table; [2] *l* is the size of *m* low marginals of dataset.

3.6.2. Conditional Probability Distribution Estimation

The conditional probability is also important for statistics. Sun et al. [92] have investigated on conditional distribution estimation for the keys in key-value data. They formalized *k*-way conditional frequency estimation and applied the advanced LDP protocols to compute *k*-way conditional distributions. Besides, Xue et al. [109] proposed to compute the conditional probability based on *k*-way marginals and further train a Bayes classifier.

3.7. Frequency Estimation on Evolving Data

So far, most academic literature focuses on frequency estimation for one-time computation with LDP. However, privacy leaks will gradually accumulate as time continues to grow under centralized DP [110–112], so does LDP [28,113]. Therefore, when applying LDP for dynamic statistics over time, an LDP-compliant method should take time factor into account. Otherwise, the mechanism is actually difficult to achieve the expected privacy protection over long time scales. For example, Tang et al. [86] have pointed out that the privacy parameters provided by Apple's implementation on MacOS will actually become unreasonably large even in relatively short time periods. Therefore, it requires careful considerations for longitudinal attacks on evolving data.

Erlingsson et al. [27] adopted a heuristic memoization technique to provide longitudinal privacy protection in the case that multiple records are collected from the same user. Their method includes Permanent randomized response and Instantaneous randomized response. These two procedures will be performed in sequence with a memoization step in between. The Permanent randomized response outputs a perturbed answer which is reused as the real answer. The Instantaneous randomized response reports the perturbed answer over time, which prevents possible tracking externalities. In particular, the longitudinal privacy protection in work [27] assumes that the user value does not change over time, such as the continual observation model [114]. Thus, the approach in [27] cannot guarantee strong privacy for the users who have numeric values with frequent changes.

Inspired by [27], Ding et al. [28] used the permanent memoization for continual counter data collection and histogram estimation. They first designed ω-bit mechanism ωBitFlip to estimate frequency of counter values in a discrete domain with *k* buckets. In ωBitFlip, each user randomly draws

ω bucket numbers without replacement from $[k]$, denoted as $j_1, j_2, \cdots, j_\omega$. At each time t, each user randomizes her data $v_t \in [k]$ and reports a vector $b_t = [(j_1, b_t(j_1)), (j_2, b_t(j_2)), \cdots, (j_\omega, b_t(j_\omega))]$, where $b_t(j_z)$ $(z = 1, 2, \cdots, \omega)$ is a random 0/1 bit with

$$\mathbb{P}[b_t(j_z) = 1] = \begin{cases} \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}, & \text{if } v_t = j_z, \\ \frac{1}{e^{\epsilon/2}+1}, & \text{if } v_t \neq j_z. \end{cases} \tag{53}$$

Assume the sum of received 1 bit is $\bar{N}_{v_t}$. Then, the estimated frequency of $v_t \in [k]$ is

$$\widehat{f_{v_t}} = \frac{k}{N\omega} \sum_{\bar{N}_{v_t}} \frac{\bar{N}_{v_t} \cdot (e^{\epsilon/2}) - 1}{e^{\epsilon/2} - 1}. \tag{54}$$

As we can see, ωBitFlip will be the same as the one in Duchi et al. [115] when $\omega = k$. Ding et al. [28] have proved that ωBitFlip holds an error bound of $O\left(\frac{\sqrt{k \log k}}{\epsilon \sqrt{N\omega}}\right)$.

In naïve memoization, each user reports a perturbed value based on the mapping $f_k : [k] \to \{0, 1\}^k$, which leads to privacy leakage. To tackle this, Ding et al. [28] proposed ω-bit permanent memoization mechanism ωBitFlipPM based on ωBitFlip. ωBitFlipPM reports each response in a mapping $f_\omega : [k] \to \{0, 1\}^\omega$, which avoids the privacy leakage since multiple buckets are mapped to the same response.

Moreover, Joseph et al. [113] proposed a novel LDP-compliant mechanism THRESH for collecting up-to-date statistics over time. The key idea of THRESH is to update the global estimation only when it might become sufficiently inaccurate. To identify these update-needed epochs, Joseph et al. designed a voting protocol that requires users to privately report a vote for whether they believe the global estimation needs to be updated. The THRESH mechanism can ensure that the privacy guarantees only degrade with the number of times of the statistics changes, rather than the number of times the computation of the statistics. Therefore, it can achieve strong privacy protection for frequency estimation over time while ensuring good accuracy.

4. Mean Value Estimation with LDP

This Section summarizes the task of mean value estimation for numeric data with LDP, including mean value estimation on numeric data and mean value estimation on evolving data.

In formal, let $\mathcal{D} = \{V^1, V^2, \cdots, V^N\}$ be the data of all users, where N is the number of users. Each tuple $V^i = (v_1^i, v_2^i, \cdots, v_d^i)$ $(i \in [1, N])$ denotes the data of the i-th user, which consists of d numeric attributes $A_1, A_2, \cdots, A_d$. Each v_j^i $(j \in [1, d])$ denotes the value of the j-th attribute of the i-th user. Without loss of generality, the domain of each numeric attribute is normalized into $[-1, 1]$. The mean estimation is to estimate the mean value of each attribute $A_j (j \in [1, d])$ over N users, i.e., $\frac{1}{N} \sum_{i=1}^{N} v_j^i$.

4.1. Mean Value Estimation on Numeric Data

Let $\widehat{V}^i = (\widehat{v}_1^i, \widehat{v}_2^i, \cdots, \widehat{v}_d^i)$ be the perturbed d-dimensional data of user i. Given a perturbation mechanism $\mathcal{M}$, we use $\mathbb{E}[\widehat{v}_j]$ to denote the expectation of the output $\widehat{v}_j$ given an input v_j. Therefore, to achieve LDP, a perturbation mechanism should satisfy the following two constraints, that is,

$$\mathbb{E}[\widehat{v}_j] = v_j, \tag{55}$$

$$\mathbb{P}[\widehat{v}_j \in \mathbb{V}|v] = 1. \tag{56}$$

The first constraint (i.e., Equation (55)) shows that the mechanism should be unbiased. The second constraint (i.e., Equation (56)) shows that the sum of probabilities of the outputs must be one, where $\mathbb{V}$ is the output range of $\mathcal{M}$.

Laplace mechanism [45] under DP can be applied in a distributed manner to achieve LDP. Based on Laplace mechanism, each user's data will be perturbed by adding randomized Laplace noise, i.e., $\widehat{V}^i = V^i + \langle Lap(\frac{d\Delta}{\epsilon}) \rangle^d$, where $Lap(\lambda)$ is a random variable drawn from a Laplace distribution with the probability density function of $pdf(v) = \frac{1}{2\lambda} \exp -\frac{|v|}{\lambda}$. Please note that Δ is the sensitivity and $\Delta = 2$ since each numeric data lies in range $[-1, 1]$. The privacy budget for each dimension is ϵ/d. Then, the aggregator will compute the average value of all received noisy reports as $\frac{1}{N} \sum_{i=1}^{N} \widehat{v}_j^i$. It can be easily verified that $\frac{1}{N} \sum_{i=1}^{N} \widehat{v}_j^i$ is an unbiased estimator of the j-th attribute since the injected Laplace noises have zero mean. Thus, the final mean value is also unbiased. Besides, the variance of the estimated $\widehat{v}_i$ is $\frac{8d^2}{\epsilon^2}$. The amount of noise of the estimated mean of each attribute is $O(\frac{d\sqrt{\log d}}{\epsilon\sqrt{N}})$, which is super-linear to dimension d. When $d = 1$, the variance is $\frac{8}{\epsilon^2}$ and the error bound is $O(\frac{1}{\epsilon\sqrt{N}})$. As we can see, the Laplace mechanism will incur excessive error when d becomes large.

Duchi et al. [115] proposed an LDP-compliant method for collecting multi-dimensional numeric data. The basic idea of Duchi et al.'s method is to use a randomized response technique to perturb each user's data according to a certain probability distribution while ensuring an unbiased estimation. Each user's tuple $V^i \in [-1, 1]^d$ will be perturbed into a noisy vector $\widehat{V}^i \in \{-B, B\}^d$, where B is a constant decided by d and ϵ. According to [115], B is computed as

$$B = \begin{cases} \frac{2^d + C_d \cdot (e^\epsilon - 1)}{\binom{d-1}{(d-1)/2} \cdot (e^\epsilon - 1)}, & \text{if } d \text{ is odd,} \\ \frac{2^d + C_d \cdot (e^\epsilon - 1)}{\binom{d-1}{d/2} \cdot (e^\epsilon - 1)}, & \text{otherwise,} \end{cases} \tag{57}$$

where

$$C_d = \begin{cases} 2^{d-1}, & \text{if } d \text{ is odd,} \\ 2^{d-1} - \frac{1}{2}\binom{d}{d/2}, & \text{otherwise.} \end{cases} \tag{58}$$

Duchi et al.'s method takes a tuple $V^i \in [-1, 1]^d$ as inputs and discretizes the d-dimensional data into $X := [X_1, X_2, \cdots, X_d] \in \{-1, 1\}^d$ by sampling each X_j independently from the following distribution.

$$\mathbb{P}[X_j = x_j] = \begin{cases} \frac{1}{2} + \frac{1}{2}v_j^i, & \text{if } x_j = 1, \\ \frac{1}{2} - \frac{1}{2}v_j^i, & \text{if } x_j = -1. \end{cases} \tag{59}$$

In the case of X is sampled, let T^+ (resp. T^-) be the set of all tuples $\widehat{V}^i \in \{-B, B\}^d$ such that $\widehat{V}^i \cdot X > 0$ (resp. $\widehat{V}^i \cdot X \leq 0$). The algorithm will return a noisy value based on the value of a Bernoulli variable u. That is, it will return $\widehat{V}^i$ uniformly at random from T^+ with probability of $\mathbb{P}[u = 1] = \frac{e^\epsilon}{e^\epsilon + 1}$ or return a noisy value $\widehat{V}^i$ uniformly at random from T^- with probability of $\mathbb{P}[u = 0] = \frac{1}{e^\epsilon + 1}$.

Duchi et al. have shown that $\frac{1}{N} \sum_{i=1}^{N} \widehat{v}_j^i$ is an unbiased estimator for each attribute A_j. Besides, the error bound of Duchi et al.'s method is $O\left(\frac{\sqrt{d \log d}}{\epsilon\sqrt{N}}\right)$.

Although Duchi et al.'s method can achieve LDP and has an asymptotic error bound, it is relatively sophisticated. Nguyên et al. [91] have pointed that Duchi et al.'s solution does not achieve ϵ-LDP when d is even. Nguyên et al. have proposed one possible solution to fix Duchi et al.'s method to satisfy LDP when d is even. Their method is to re-define a Bernoulli variable u such that

$$\mathbb{P}[u = 1] = \frac{e^\epsilon \cdot C_d}{(e^\epsilon - 1)C_d + 2^d}. \tag{60}$$

Furthermore, Nguyên et al. proposed Harmony [91] that is simpler than Duchi et al.'s method when collecting multi-dimensional data with LDP, but achieves the same privacy guarantee and

asymptotic error bound. Given an input V^i, Harmony returns a perturbed tuple $\widehat{V}^i$ which has non-zero value on only one dimension $j \in [1, d]$. That is, Harmony uniformly at random samples only one dimension j from $[1, d]$ and returns a noisy value $\widehat{v}^i_j$ that is generated from the following distribution

$$\mathbb{P}\left[\widehat{v}^i_j = x\right] = \begin{cases} \frac{v^i_j \cdot (e^\epsilon - 1) + e^\epsilon + 1}{2(e^\epsilon + 1)}, & \text{if } x = \frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d, \\ \frac{-v^i_j \cdot (e^\epsilon - 1) + e^\epsilon + 1}{2(e^\epsilon + 1)}, & \text{if } x = -\frac{e^\epsilon + 1}{e^\epsilon - 1} \cdot d. \end{cases} \tag{61}$$

As we can see, in Harmony, each user only needs to report one bit to the aggregator. Thus, Harmony has a lower communication overhead of $O(1)$ than Duchi et al.'s method, but holds the same error bound of $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{N}}\right)$.

Wang et al. [29] further proposed piecewise mechanism (PM) that has lower variance and is easier to implement than Duchi et al.'s method.

We first introduce PM for one dimensional data, i.e., $d = 1$. Given an input $v^i \in [-1, 1]$ of user i, the PM outputs a perturbed value $\widehat{v}^i$ in $[-C, C]$, where $C = \frac{e^{(\epsilon/2)} + 1}{e^{(\epsilon/2)} - 1}$. The probability density function (pdf) of $\widehat{v}^i$ follows a piecewise constant function as

$$pdf(\widehat{v}^i = x | v^i) = \begin{cases} p, & \text{if } x \in [\ell_{v^i}, r_{v^i}], \\ \frac{p}{e^\epsilon}, & \text{if } x \in [-C, \ell_{v^i}] \cup [r_{v^i}, C], \end{cases} \tag{62}$$

where $p = \frac{e^\epsilon - e^{\epsilon/2}}{2(e^{\epsilon/2} + 1)}$, $\ell_{v^i} = (C + 1)/2 \cdot v^i - (C - 1)/2$, and $r_{v^i} = \ell_{v^i} + C - 1$.

Based on Equation (62), PM samples a value u uniformly at random from $[0, 1]$ and returns $\widehat{v}^i$ uniformly at random from $[\ell_{v^i}, r_{v^i}]$ if $u < \frac{e^{\epsilon/2}}{e^{\epsilon/2} + 1}$ (or returns $\widehat{v}^i$ uniformly at random from $[-C, \ell_{v^i}] \cup [r_{v^i}, C]$ if $\frac{e^{\epsilon/2}}{e^{\epsilon/2} + 1} \le u \le 1$).

Wang et al. [29] have proved that the variance of PM for one-dimensional data is $\frac{4e^{\epsilon/2}}{3(e^{\epsilon/2} - 1)^2}$. Recall that the variance of Duchi et al.'s method for one-dimensional data is $\frac{(e^\epsilon + 1)^2}{(e^\epsilon - 1)^2}$. It can be verified that the variance of PM will smaller than that of Duchi et al.'s method when $\epsilon > 1.29$.

Furthermore, Wang et al. [29] extended the PM for collecting multi-dimensional data based on the idea of Harmony [91]. Given an input tuple $V^i \in [-1, 1]^d$, it returns a perturbed $\widehat{V}^i$ that has non-zero value on k dimensions at most, where $m = \max\{1, \min\{d, \lfloor \frac{\epsilon}{2.5} \rfloor\}\}$. In this way, the PM for multi-dimensional data has a error bound of $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{N}}\right)$. In particular, m is much smaller than d and equals to 1 for $\epsilon < 5$.

Comparisons and discussions. Table 9 summarizes of LDP algorithms for mean estimation on multi-dimensional numeric data. As we can see, Laplace [45] and Duchi et al.'s method cost high communication overhead while Harmony [91] and PM [29] have low communication costs. For d-dimensional data, Laplace has the largest error bound. In contrast, the other three mechanisms have lower error bounds than Laplace. Please note that, in theory, PM [29] holds the communication cost of $O(k)$ and error bound of $O\left(\frac{\sqrt{dk \log d}}{\epsilon \sqrt{N}}\right)$, where k is much smaller than d and will be 1 when $\epsilon < 5$.

The last column of Table 9 shows the variance of each mechanism for one-dimensional data. It can be verified that the variance of PM is always smaller than Laplace, but slightly worse than Duchi et al.'s method and Harmony when $\epsilon < 1.29$. By observing this, Wang et al. [29] proposed to combine PM and Duchi et al.'s method into a new Hybrid Mechanism (HM). They have proved that the worst-case variance of HM for $d = 1$ is

$$Var_{HM} = \begin{cases} \frac{e^{\epsilon/2}+3}{3e^{\epsilon/2}(e^{\epsilon/2}-1)} + \frac{(e^{\epsilon}+1)^2}{e^{\epsilon/2}(e^{\epsilon}-1)^2}, & \text{for } \epsilon > 0.61, \\ \frac{(e^{\epsilon}+1)^2}{(e^{\epsilon}-1)^2}, & \text{for } \epsilon \le 0.61. \end{cases} \tag{63}$$

Thus, it can be verified the variance of HM is always smaller than other mechanisms in Table 9.

Table 9. Comparisons of mean estimation mechanisms on d-dimensional numeric data with LDP.

Algorithms	Comm. Cost	Error Bound	Variance ($d = 1$)
Laplace [45]	$O(d)$	$O\left(\frac{d\sqrt{\log d}}{\epsilon\sqrt{N}}\right)$	$\frac{8}{\epsilon^2}$
Duchi et al. [115]	$O(d)$	$O\left(\frac{\sqrt{d\log d}}{\epsilon\sqrt{N}}\right)$	$\frac{(e^{\epsilon}+1)^2}{(e^{\epsilon}-1)^2}$
Harmony [91]	$O(1)$	$O\left(\frac{\sqrt{d\log d}}{\epsilon\sqrt{N}}\right)$	$\frac{(e^{\epsilon}+1)^2}{(e^{\epsilon}-1)^2}$
PM [29]	$O(m)$	$O\left(\frac{\sqrt{d\log d}}{\epsilon\sqrt{N}}\right)$	$\frac{4e^{\epsilon/2}}{3(e^{\epsilon/2}-1)^2}$
HM [29]	$O(m)$	$O\left(\frac{\sqrt{d\log d}}{\epsilon\sqrt{N}}\right)$	Equation (63)

4.2. Mean Value Estimation on Evolving Data

As pointed in Section 3.7, the privacy leakage will accumulate with the increase of time. This also exists in the mean estimation. Ding et al. [28] employed both α-point rounding and memoization techniques to estimate mean value of the counter data while ensuring strong privacy protection over time. The basic idea of α-point rounding is to discretize the data domain based on a discretization granularity s. Ding et al. proposed 1-bit mechanism 1BitMean for mean estimation. Assume that each user i has a private value $v^i(t) \in [0, r]$ at time t. The 1BitMean requires that each user reports one bit $b^i(t)$ that is drawn from the distribution

$$b^i(t) = \begin{cases} 1, & \text{with probability } \frac{1}{e^{\epsilon}+1} + \frac{v^i(t)}{r} \cdot \frac{e^{\epsilon}-1}{e^{\epsilon}+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{64}$$

The mean value of N users at time t can be estimated as

$$\widehat{m}(t) = \frac{r}{N} \sum_{i=1}^{N} \frac{b^i(t) \cdot (e^{\epsilon}+1) - 1}{e^{\epsilon}-1}. \tag{65}$$

Based on 1BitMean, the procedure of α-point rounding includes the following four steps. (i) Discretizes data domain r into s parts. (ii) Each user i randomly picks a value $\alpha^i \in \{0, 1, \cdots, s-1\}$. (iii) Each user computes and memoize 1-bit response by invoking 1BitMean. (iv) Each user performs α-rounding based on an arithmetic progression that rounds value to the left is $v^i + \alpha^i < R$, otherwise rounds value to the right. Ding et al. [28] have proved that the accuracy of α-point rounding mechanism is the same as 1BitMean and is independent of the choice of discretization granularity s.

5. Machine Learning with LDP

Machine learning, as an essential data analysis method, was applied to various fields. However, the training process may be vulnerable to many attacks (such as membership inference attacks [116], memorizing model attacks [117], model inversion attacks [118]). For example, adversaries may extract the memorized information in the training process to approximate the sensitive data of the users [117]. Even worse is that Fredrikson et al. [118] have shown an example that the adversary could recover

images from a facial recognition system under model inversion attacks, which shows the weakness of a trained machine learning model.

The machine learning algorithms with global DP were extensively studied by imposing private training [119–123]. With the introduction of LDP, the machine learning algorithms with LDP were also investigated to achieve privacy protection in a distributed way. The following subsections summarize the existing machine learning algorithms with LDP from the perspective of supervised learning, unsupervised learning, empirical risk minimization, deep learning, reinforcement learning, and federated learning.

5.1. Supervised Learning

Supervised learning algorithms focus on training a prediction model describing data classes via a set of labeled datasets.

Yilmaz et al. [124] proposed to train a Naïve Bayes classifier with LDP. Naïve Bayes classification is to find the most probable label when given a new instance. In order to compute the conditional distributions, we need to keep the relationships between the feature values and class labels when perturbing input data. To keep this relationship, Yilmaz et al. transformed each user's value and label into a new value first and then performed LDP perturbation. Xue et al. [109] also aimed at training a Naïve Bayes classifier with LDP. They proposed to leverage the joint distributions to compute the conditional distributions. Besides, Berrett and Butucea [125] further considered the binary classification problem with LDP.

High-dimensionality is a big challenge for training a classifier with LDP, which will result in huge time cost and low accuracy. One of the traditional solutions is dimensionality reduction, such as Principal Component Analysis (PCA) [126]. However, the effective dimensionality reduction methods with LDP in machine learning still need further research. Moreover, user partition is always used when learning a model with LDP. For example, the work in [124] partitions users into two groups to compute the mean value and squares, respectively. However, simply partitioning users into groups will reduce the estimation accuracy. Therefore, research on supervised learning with LDP still has a long way to go.

5.2. Unsupervised Learning

The problem of clustering was studied under centralized DP [127–129]. With LDP model, Nissim and Stemmer [130] conducted 1-clustering by finding a minimum enclosing ball. Moreover, Sun et al. [131] have investigated the non-interactive clustering under LDP. They extended the Bit Vector mechanism in [132,133] by modifying the encoding process and proposed kCluster algorithm in an anonymous space based on the improved encoding process. Furthermore, Li et al. [134] proposed a local-clustering-based collaborative filtering mechanism that uses the kNN algorithm to group items and ensure the item-level privacy specification.

For clustering in the local model, the respondent randomizes her/his own data and reports to an untrusted data curator. Although the accuracy of local clustering is not good as that in the central model, local clustering algorithms can achieve stronger privacy protection for users and are more practical for privacy specifications, such as personalized privacy parameters [57,135]. Xia et al. [136] applied LDP on *K*-means clustering by directly perturbing the data of each user. They proposed a budget allocation scheme to reduce the scale of noise to improve accuracy. However, the investigation on clustering under LDP is still in the early stage of research.

5.3. Empirical Risk Minimization

In machine learning, the error computed from training data is called empirical risk. Empirical risk minimization (ERM) is such a process that computes an optimal model from a set of parameters by minimizing the expected loss [137]. A loss function $\mathcal{L}(\theta; x, y)$ is parameterized by x, y and aims to

map the parameter vector θ into a real number. The goal of ERM is to identify a parameter vector θ^* such that

$$\theta^* = \arg\min_{\theta} \left[\frac{1}{N} \left(\sum_{i=1}^{N} \mathcal{L}(\theta; x_i, y_i) \right) + \frac{\lambda}{2} \|\theta\|_2^2 \right], \tag{66}$$

where $\lambda > 0$ is the regularization parameter.

By choosing different loss functions, ERM can be used to solve certain learning tasks, such as logistic regression, linear regression, and support vector machine (SVM). The interactive model and non-interactive model under LDP were discussed in the existing literature for natural learning problems [32,49]. Apparently, the interactive model has a better accuracy but leads to high network delay and week privacy guarantee. The non-interactive model is strictly stronger and more practical in most settings.

Smith et al. [138] initiated the investigation of interaction in LDP for natural learning problems. They pointed out that for a large class of convex optimization problems with LDP, the server needs to exchange information with each user back and forth in sequence, which will lead to network delays. Thus, they investigated the necessity of the interactivity to optimize convex functions. Smith et al. also provided new algorithms that are either non-interactive or only use a few rounds of interaction. Moreover, Zheng et al. [139] further proposed more efficient algorithms based on Chebyshev expansion under non-interactive LDP, which achieves quasi-polynomial sample complexity bound.

However, the sample complexity in [138,139] is exponential with the dimensionality and will become not very meaningful in high dimensions. In fact, it is quite common to involve high dimensions in machine learning. Wang et al. [32] have proposed LDP algorithms with its error bound depending on the Gaussian width, which improves the one in [138], but the sample complexity still is exponential with the dimensionality. Their following work in [140] improves the sample complexity to be quasi-polynomial. However, the practical performance of these algorithms is still limited. Therefore, for the generalized linear model (GLM), Wang et al. [141] further proved that when the feature vector of GLM is sub-Gaussian with bounded ℓ_1-norm, then the LDP algorithm for GLM will achieve a fully polynomial sample complexity. Furthermore, Wang and Xu [142] addressed the principal component analysis (PCA) problem under the non-interactive LDP and proved the lower bound of the minimax risk in both the low and high dimensional settings.

Moreover, the works in [29,91] built the classical machine learning models under LDP in a way of empirical risk minimization (ERM). They solved ERM by stochastic gradient descent (SGD). They consider three common machine learning models: linear regression, logistic regression and SVM classification. The SGD algorithm is adopted to compute the target parameter θ that holds the minimum (or hopefully) loss. Specifically, at each iteration $t + 1$, the parameter vector is updated as $\theta_{t+1} = \theta_t - \eta \cdot \nabla \mathcal{L}(\theta_t; x, y)$, where η is the learning rate, $\langle x, y \rangle$ is a tuple of randomly selected user, $\nabla \mathcal{L}(\theta_t; x, y)$ is the gradient of loss function $\mathcal{L}(\theta_t)$ at θ_t. With LDP, each $\nabla \mathcal{L}$ is perturbed into a noisy gradient $\nabla \mathcal{L}^*$ by an LDP-compliant algorithm before reporting to the aggregator. That is,

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{1}{|G|} \sum_{i \in G} \nabla \mathcal{L}_i^*, \tag{67}$$

where $\nabla \mathcal{L}_i^*$ is the perturbed gradient of the user u_i and $|G|$ is the batch size.

When considering the data center network (DCN), Fan et al. [143] investigated the LDP-based support vector regression (SVR) classification for cloud computing supported data centers. Their method achieves LDP based on the Laplace mechanism. Similarly but differently, Yin et al. [144] studied the LDP-based logistic regression classification by involving three specific steps, i.e., noise addition, feature selection, and logistic regression. LDP is also applied to online convex optimization to avoid disclosing any parameters while realizing unconstrained adaptive online learning [145]. Besides, Jun and Orabona [146] studied the parameter-free SGD problem under LDP. They proposed BANCO

that achieves the convergence rate of the tuned SGD without repeated runs, thus reducing privacy loss and saving the privacy budget.

5.4. Deep Learning

Deep learning has played an important role in natural language processing, image classification, and so on. However, the adversaries can easily inject the malicious algorithms in the training process, and then extract and approximate the sensitive information of users [117,147].

Arachchige et al. [34] proposed an LDP-compliant mechanism LATENT to control privacy leaks in deep learning models (i.e., convolutional neural network (CNN)). Just like other LDP frameworks, LATENT integrates a randomization layer (i.e., LDP layer) to against the untrusted learning servers. A big challenge of applying LDP in deep learning is that the sensitivity is extremely large. In LATENT, the sensitivity is lr, where l is the length of the binary string and r is the number of layers of the neural network. To address this, Arachchige et al. further improved OUE [35] where the sensitivity is 2. They proposed the modified OUE (MOUE) that has more flexibility for controlling the randomization of 1s and increasing the probability of keeping 0 bits in their original state. Besides, Arachchige et al. proposed utility enhancing randomization (UER) mechanism that further improves the utility of the randomized binary strings.

Furthermore, by using the teacher-student paradigm, Zhao in [148] investigated the distributed deep learning model under DP and further considered to allow the personalized choice of privacy parameters for each distributed data entity under LDP. Xu et al. [149] have applied LDP on a deep inference-based edge computing framework to privately build complex deep learning models. Overall, deep learning with LDP is in the early stage of research. Further research is still needed to provide strong privacy, reduce high dimensionality, and improve accuracy.

5.5. Reinforcement Learning

Reinforcement learning enables an agent to learn a model interactively, which was widely adopted in artificial intelligence (AI). However, reinforcement learning is vulnerable to potential attacks, leading to serious privacy leakages [150].

To protect user privacy, Gajane et al. [151] initially studied the multi-armed bandits (MAB) problem with LDP. They proposed a bandit algorithm with LDP aiming at arms with Bernoulli rewards. Afterward, Basu et al. [152] proposed a unifying set of fundamental privacy definitions for MAB algorithms with the graphical model and LDP model. They have provided both the distribution-dependent and distribution-free regret lower bounds.

As for distributed reinforcement learning, Ono and Takahashi [153] proposed a framework Private Gradient Collection (PGC) to privately learn a model based on the noisy gradients. Under the PGC, each local agent reports the perturbed gradients that satisfy LDP to the central aggregator who will update the global parameters. Besides, Ren et al. [154] mainly investigated on the regret minimization for MAB problems with LDP and proved a tight regret lower bound. They proposed two algorithms that achieve LDP based on Laplace perturbation and Bernoulli response, respectively.

Reinforcement learning plays an important role in AI [155]. LDP is a potential technique to prevent sensitive information from leakage in reinforcement learning. However, LDP-based reinforcement learning is still in its infancy.

5.6. Federated Learning

Federated learning (FL) is one of the core technologies for the development of a new generation of artificial intelligence (AI) [156–158]. It provides attractive collaborative learning frameworks for multiple data owners/parties [159]. Although FL itself can effectively balance the trade-off between utility and privacy for machine learning [160], serious privacy issues still occurred when transmitting or exchanging model parameters. Therefore, LDP was widely adopted in FL systems to provide strong privacy guarantees, such as in smart electric power systems [161] or wireless channels [162].

The studies in [163,164] have adopted global DP [22] to protect sensitive information in FL. However, since FL itself is a distributed learning framework, LDP is more appropriate to FL systems. Truex et al. [165] integrated LDP into a FL system for joint training of deep neural networks. Their method can efficiently handle complex models and against inference attacks while achieving personalized LDP. Besides, Wang et al. [33] proposed FedLDA that is an LDP-based latent Dirichlet allocation (LDA) model in the setting of FL. FedLDA adopts a novel random response with prior, which ensures that the privacy budget is irrelevant to the dictionary size and the accuracy is greatly improved by an adaptive and non-uniform sampling processing.

To improve the model-fitting and prediction of the schemes, Bhowmick et al. [166] proposed a relaxed optimal LDP mechanism for private FL. Li et al. [167] introduced an efficient LDP algorithm for meta-learning, which can be applied to realize the personalized FL. As for federated SGD, LDP was adopted to prevent privacy leakages from gradients. However, the increase of dimension d will cause the privacy budget to decay rapidly and the noise scale to increase, which leads to poor accuracy of the learned model when d is large. Thus, Liu et al. [168] proposed FedSel that only selects the most important top-k dimensions under the premise of stabilizing the learning process. In addition, Sun et al. [169] proposed to mitigate the privacy degradation by splitting and shuffling, which reduces noise variance and improve accuracy.

Recently, Naseri et al. [170] proposed an analytical framework that empirically assesses the feasibility and effectiveness of LDP and CDP in protecting FL. They have shown that both LDP and global DP can defend against backdoor attacks, but not do well for property inference attacks.

6. Applications

This Section concludes the wide applications of LDP for real practice and the Internet of Things.

6.1. LDP in Real Practice

LDP was applied to many real systems due to its powerfulness in privacy protection. There are several large scale deployments in the industry as follows.

- RAPPOR in Google Chrome. As the first practical deployment of LDP, RAPPOR [27] is proposed by Google in 2014 and was integrated into Google Chrome to constantly collect the statistics of Chrome usage (e.g., the homepage and search engine settings) while protecting users' privacy. By analyzing the distribution of these settings, the malicious software tampered with the settings without user consent will be targeted. Furthermore, a follow-up work [87] by the Google team has extended RAPPOR to collect more complex statistical tasks when there is no prior knowledge about dictionary knowledge.

- LDP in iOS and MacOS. The Apple company has deployed LDP in iOS and MacOS to collect typing statistics (e.g., emoji frequency detection) while providing privacy guarantees for users [26,85,171]. The deployed algorithm uses the Fourier transformation and sketching technique to realize a good trade-off between massive learning and high utility.

- LDP in Microsoft Systems. Microsoft company has also adopted LDP and deployed LDP starting with Windows Insiders in Windows 10 Fall Creators Update [28]. This deployment is used to collect telemetry data statistics (e.g., histogram and mean estimations) across millions of devices over time. Both α-point rounding and memoization technique are used to solve the problem that privacy leakage accumulates as time continues to grow.

- LDP in SAP HANA. SAP announced that its database management system SAP HANA has integrates LDP to provide privacy-enhanced processing capabilities [172] since May 2018. There are two reasons for choosing LDP. One is to provide privacy guarantees when counting sum average on the database. The other is to ensure a maximum transparency of the privacy-enhancing methods since LDP can avoid the trouble and overhead of maintaining a privacy budget.

Many other companies (e.g., Firefox [84] and Samsung [91]) also plan to build LDP-compliant systems to collect the usage statistics of users while providing strong privacy guarantees. It still needs great efforts to research on large-scale, efficient, and accurate privacy-preserving frameworks while monitoring the behaviors of client devices.

6.2. LDP in Various Fields

With the rapid development of the Internet of Things (IoT), multiple IoT applications generate big multimedia data that relate to user health, traffic, city surveillance, locations, etc. [173]. These data are collected, aggregated, and analyzed to facilitate the IoT infrastructures. However, privacy leakages of users hindered the development of IoT systems. Therefore, LDP plays an important role in data privacy protection in the Internet of Things.

Usman et al. [174] proposed a privacy-preserving framework PAAL by combining authentication and aggregation with LDP. PAAL can provide each end-device user with strong privacy guarantee by perturbing the aggregated sensitive information. Ou et al. [175] also adopted LDP to prevent the adversary from inferring the time-series data classification of household appliances. As a promising branch of IoT, the Internet of Vehicles (IoV) has stimulated the development of vehicular crowdsourcing applications, which also results in unexpected privacy threats on vehicle users. Zhao et al. [176] adopted both LDP and FL models to avoid sensitive information leakage in IoV applications. Besides, the work in [39] has provided a detailed summary of the applications of LDP in the Internet of connected vehicles.

In what follows, we summarize more specific applications of LDP in the Internet of Things.

6.2.1. Edge Computing

LDP itself is a distributed privacy model that can be easily used for providing strict privacy guarantees in edge computing applications. Xu et al. [149] proposed a lightweight edge computing framework based on deep inference while achieving LDP for mobile data analysis. Moreover, Song et al. [177] also leveraged LDP models to protect the privacy of multi-attribute data based on edge computing. They solved the problem of maximizing data utility under privacy budget constrains, which improves the accuracy greatly.

6.2.2. Hypothesis Testing

Many existing studies looked at the intersection of DP and hypotheses testing [178–181]. The private hypothesis testing under LDP has also been studied in [115,182–186], including identity and independence testing, Z-test, and distribution testing.

Duchi et al. [115] initiated to define the canonical hypothesis testing problem and studied the error bound of the probability of error in the hypothesis testing problem. From the perspective of information theory, Kairouz et al. [63,182] studied the maximizing of f-divergence utility functions under the constraints of LDP and made the effective sample size reduce from N to $\epsilon^2 N$ for hypothesis testing.

Both studies in [183,184] investigated hypothesis testing with LDP and presented the asymptotic power and the sample complexity. Sheffet in [183] showed a characterization for hypothesis testing with LDP and proved the bound of sample complexities for both identity-testing and independence testing when using randomized response techniques. Gaboardi and Rogers [184] focused on the goodness of fit and independence hypothesis tests under LDP. They designed three different goodness of git tests that use different protocols to achieve LDP and guarantee the convergence to a chi-square distribution. Afterward, Gaboardi et al. [185] further provided upper- and lower-bounds for mean estimation under (ϵ, δ)-LDP and showed the performance of LDP-compliant Z-test algorithm. Moreover, Acharya et al. in [186] presented the optimal locally private distribution testing algorithm with optimal sample complexity, which improves on the sample complexity bounds in [183].

6.2.3. Social Network

In the Internet of Things, social network analysis has drawn much attention from many parties. Providing users with effective privacy guarantees is a key prerequisite for analyzing social network data. Therefore, privacy-preserving on social network publishing was widely investigated with LDP. Usually, a social network is formalized as a graph.

A big challenge is how to apply LDP on the aggregation and generation of the complex graph structures. Qin et al. [187] initially formulated the problem of generating synthetic decentralized social graphs with LDP. In order to obtain graph information from simple statistics, they proposed LDPGen that incrementally identifies clusters of connected nodes under LDP to capture the structure of the original graph. Zhang et al. [188] adopted the idea of multi-party computation clustering to generate a graph model under the optimized RR algorithm. Besides, Liu et al. [189] used the perturbed local communities to generate a synthetic network that maintains the original structural information.

The studies in [190,191] focus on online social networks (OSN) publishing with LDP. The key idea of [190,191] is graph segmentation, which can reduce the noise scale and the output space. Specifically, they split the original graph to generating the representative hierarchical random graphs (HRG) and then perturbed each local graph with LDP. Yang et al. [192] also applied a hierarchical random graph model for publishing hierarchical social networks with LDP. They further employed the Monte Carlo Markov chain to reduce the possible output space and improve the efficiency while extracting HRG with LDP. In addition, Ye et al. [193] focused on building an LDP-enabled graph metric estimation framework that supports a variety of graph analysis tasks, such as graph clustering, synthetic graph generation, community detection, etc. They proposed to compute the most popular graph metrics only by three parameters, i.e., adjacency bit vector, adjacency matrix, and node degree.

Furthermore, social networks are decentralized in nature. In addition to containing her own connections, a participant's local view also contains the connections of her neighbors that are private for the neighbors, but not directly private for the participant herself. In this case, Sun et al. [194] indicated that the general LDP is insufficient to protect the privacy of all participants. Therefore, they formulated a stringent definition of decentralized differential privacy (DDP) that ensures the privacy of both participant and her neighbors. Besides, Wei et al. [195] further proposed AsgLDP for generating decentralized attributed graphs with LDP. Specifically, AsgLDP is composed of two phases that are used for unbiased information aggregation and attributed graph generation, respectively, which preserves the important properties of social graphs.

As for social network analysis, it is hard to obtain a global view of networks since the local view of each user is much limited. The most existing methods solve this challenge by partitioning users into disjoint groups (e.g., [187,188]). However, the performance of this approach is severely restricted by the number of users. Besides, the computational cost should also be considered for large scale graphs.

6.2.4. Recommendation System

The recommendation system makes most of the services and applications in the Internet of Things feasible [196]. However, the recommendation system may abuse user data and extract private information when collecting the rating pairs from each user [197].

While integrating the privacy-preserving techniques into the recommendation system, a big challenge is how to compromise privacy and usability. Liu et al. [198] proposed an unobtrusive recommendation system that balances between privacy and usability by crowdsourcing user privacy settings and generating corresponding recommendations. To provide stronger privacy guarantees, Shin et al. [199] proposed LDP-based matrix factorization algorithms that protect both user's items and ratings. Meanwhile, they used the dimensionality reduction technique to reduce domain space, which improves the data utility. Jiang et al. [200] proposed a more reliable Secure Distributed Collaborative Filtering (SDCF) framework that ables to preserve the privacy of data items, recommendation model, and the existence of the ratings at the same time. Nonetheless, SDCF performs RAPPOR in each iteration to achieve LDP protection, which will lead to a large

perturbation error and low accuracy. To ensure the accuracy, Guo et al. [201] proposed to reconstruct the collaborative filtering based on similarity scores, which greatly improves the trade-off between privacy and utility.

Although many studies investigated the recommendation system with LDP, the low accuracy cased by high dimensionality and high sparse rating dataset still remains a huge challenge [202]. Besides, the targeted LDP protocols for recommendation systems need further study.

7. Discussions and Future Directions

This section presents some discussions and research directions for LDP.

7.1. Strengthen Theoretical Underpinnings

There are several theoretical limitations under LDP needing to be further strengthened.

(1) The lower bound of the accuracy is not very clear. Duchi et al. [115] showed a provably optimal estimation procedure under LDP. However, the lower bounds on the accuracy of other LDP protocols should be proved elaborately.

(2) Can the sample complexity under LDP be further reduced? One of the limitations of LDP algorithms is that the number of users should be substantially large to ensure the data utility. A general rule of thumb [27] is $\prod_{i=1}^{d} |\Omega_i| \propto \sqrt{N}/10$, where $|\Omega_i|$ is the domain size of the ith attribute and N is the data size. Some studies in [56,67] have focused on the scenarios with a small number of users and tried to reduce the sample complexity for all privacy regimes.

(3) There is relatively little research on relaxation of LDP (i.e., (ϵ, δ)-LDP). It needs to be theoretically studied to show whether we can get any improvements in utility or other factors when adding a relaxation to LDP [81].

(4) The more general variant definitions of LDP can be further studied. Some novel and more strict definitions based on LDP were proposed. However, they are only for some specific datasets. For example, d-privacy is only for location datasets, and decentralized differential privacy (DDP) is only for graph datasets.

7.2. Overcome the Challenge of Knowing Data Domain.

As shown in Section 3, most classical LDP-based frequency estimation mechanisms need to know the domain of attributes in advance, such as PrivSet [76], LDPMiner [30]. However, it is unreasonable and hard to make an assumption about the attribute domain. For example, when estimating the statistics of input words of users, the domain of words is very huge and the new word may occur over time. Both studies in [87,88] try to address this problem. However, the error bounds still depend on the size of character domain and the node number of a trie, which limits the data utility when data domain is large. Thus, how to overcome the challenge of knowing data domain remains challenging.

7.3. Focus on Data Correlations

One of the limitations of existing methods is the neglect of data correlations. Such data correlations appear in multiple attributes [105], repeatedly collected data [27], or evolving data [28]. These correlations will leak some additional information about users. However, many LDP mechanisms neglect such inadvertent correlations, leading to a degradation of privacy guarantees. Some studies [90,93] focused on the correlations of key-value data. The research in [28,113] focused on the temporal correlations of evolving data. However, it still remains open problems of how to learn the data correlations privately and how to integrate such correlations into the encoding principles of LDP protocols.

7.4. Address High-Dimensional Data Analysis

The privacy-preserving data analysis on high-dimensional data always suffers from high computation/communication costs and low data utility, which is manifested in two aspects when using LDP. The first is computing joint probability distributions [72,105] (or, *k*-way marginals [31]). In this case, the total domain size increases exponentially, which leads to huge computing costs and low data utility due to the "curse of dimensionality". The second is protecting high-dimensional parameters of learning models in machine learning, deep learning, or federated learning tasks [168]. In this case, the scale of injected noise is proportional to the dimension, which will inject heavier noise and result in inaccurate models. Therefore, addressing high-dimensional data analysis is an urgent concern in the future.

7.5. Adopt Personalized/Granular Privacy Constraints

Since different users or parties may have distinct privacy requirements, it is more appropriate to design personalized or granular LDP algorithms to protect the data with distinct sensitivity levels. LDP itself is a distributed privacy notion. It can easily achieve personalized/granular privacy protection. Some existing work [57,203] aimed to propose personalized LDP-based frameworks for private histogram estimation. Gu et al. [59] presented Input-Discriminative LDP (ID-LDP) that is a fine-grained privacy notion and reflects the distinct privacy requirements of different inputs. However, adopting personalized/granular privacy constraints still raises further concerns when considering complex system architectures and ensuring good data utility.

7.6. Develop Prototypical Systems

LDP was widely adopted to deal with many analytic tasks and implemented in many real applications, such as Google Chrome [27], federated learning systems [165]. However, the prototypical systems based on LDP hardly appeared at present. By developing prototypical systems, we can further improve the LDP algorithms based on user requirements and the running results in an interactive way.

8. Conclusions

Data statistics and analysis have greatly facilitated the progress and development of the information society in the Internet of Things. As a strong privacy model, LDP was widely adopted to protect users against information leaks while collecting and analyzing users' data. This paper presents a comprehensive review of LDP, including privacy models, data statistics and analysis tasks, enabling mechanisms, and applications. We systematically categorize the data statistics and analysis tasks into three aspects: frequency estimation, mean estimation, and machine learning. For each category, we summarize and compare the state-of-the-art LDP-based mechanisms from different perspectives. Meanwhile, several applications in real systems and the Internet of Things are presented to demonstrate how LDP to be implemented in real-world scenarios. At last, we explore and conclude some future research directions from several perspectives.

Author Contributions: Conceptualization, T.W.; methodology, T.W.; formal analysis, T.W.; investigation, T.W.; resources, T.W., X.Z.; writing—original draft preparation, T.W.; writing—review and editing, X.Z., J.F. and X.Y. All authors have read and agreed to the published version of the manuscript.

References

1. Cheng, X.; Fang, L.; Yang, L.; Cui, S. Mobile Big Data: The Fuel for Data-Driven Wireless. *IEEE Internet Things J.* **2017**, *4*, 1489–1516. [CrossRef]

2. Guo, B.; Wang, Z.; Yu, Z.; Wang, Y.; Yen, N.Y.; Huang, R.; Zhou, X. Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm. *ACM Comput. Surv.* **2015**, *48*, 1–31. [CrossRef]

3. Shu, J.; Jia, X.; Yang, K.; Wang, H. Privacy-Preserving Task Recommendation Services for Crowdsourcing. *IEEE Trans. Services Comput.* **2018**, 1–13. [CrossRef]

4. Lu, R.; Jin, X.; Zhang, S.; Qiu, M.; Wu, X. A Study on Big Knowledge and Its Engineering Issues. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 1630–1644. [CrossRef]

5. Jarrett, J.; Blake, M.B.; Saleh, I. Crowdsourcing, Mixed Elastic Systems and Human-Enhanced Computing— A Survey. *IEEE Trans. Serv. Comput.* **2018**, *11*, 202–214. [CrossRef]

6. Krzywicki, A.; Wobcke, W.; Kim, Y.S.; Cai, X.; Bain, M.; Mahidadia, A.; Compton, P. Collaborative filtering for people-to-people recommendation in online dating: Data analysis and user trial. *Int. J. Hum.-Comput. Stud.* **2015**, *76*, 50–66. [CrossRef]

7. Chen, R.; Li, H.; Qin, A.K.; Kasiviswanathan, S.P.; Jin, H. Private spatial data aggregation in the local setting. In Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland, 16–20 May 2016; pp. 289–300.

8. Yao, Y.; Xiong, S.; Qi, H.; Liu, Y.; Tolbert, L.M.; Cao, Q. Efficient histogram estimation for smart grid data processing with the loglog-Bloom-filter. *IEEE Trans. Smart Grid* **2014**, *6*, 199–208. [CrossRef]

9. Liu, Y.; Guo, W.; Fan, C.I.; Chang, L.; Cheng, C. A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1767–1774. [CrossRef]

10. Fung, B.C.; Wang, K.; Chen, R.; Yu, P.S. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.* **2010**, *42*, 1–53. [CrossRef]

11. Zhu, T.; Li, G.; Zhou, W.; Yu, P.S. Differentially Private Data Publishing and Analysis: A Survey. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1619–1638. [CrossRef]

12. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A survey on security and privacy issues in Internet-of-Things. *IEEE Internet Things J.* **2017**, *4*, 1250–1258. [CrossRef]

13. Soria-Comas, J.; Domingo-Ferrer, J. Big data privacy: Challenges to privacy principles and models. *Data Sci. Eng.* **2016**, *1*, 21–28. [CrossRef]

14. Yu, S. Big privacy: Challenges and opportunities of privacy study in the age of big data. *IEEE Access* **2016**, *4*, 2751–2763. [CrossRef]

15. Sun, Z.; Strang, K.D.; Pambel, F. Privacy and security in the big data paradigm. *J. Comput. Inf. Syst.* **2020**, *60*, 146–155. [CrossRef]

16. Hino, H.; Shen, H.; Murata, N.; Wakao, S.; Hayashi, Y. A versatile clustering method for electricity consumption pattern analysis in households. *IEEE Trans. Smart Grid* **2013**, *4*, 1048–1057. [CrossRef]

17. Zhao, J.; Jung, T.; Wang, Y.; Li, X. Achieving differential privacy of data disclosure in the smart grid. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 504–512.

18. Barbosa, P.; Brito, A.; Almeida, H. A technique to provide differential privacy for appliance usage in smart metering. *Inf. Sci.* **2016**, *370*, 355–367. [CrossRef]

19. Wang, T.; Zhao, J.; Yu, H.; Liu, J.; Yang, X.; Ren, X.; Shi, S. Privacy-preserving Crowd-guided AI Decision-making in Ethical Dilemmas. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1311–1320.

20. General Data Protection Regulation GDPR. Available online: https://gdpr-info.eu/ (accessed on 25 May 2018).

21. Privacy Framework. Available online: https://www.nist.gov/privacy-framework (accessed on 11 July 2019).

22. Dwork, C.; Roth, A.; others. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]

23. Yang, X.; Wang, T.; Ren, X.; Yu, W. Survey on improving data utility in differentially private sequential data publishing. *IEEE Trans. Big Data* **2017**, 1–19. [CrossRef]

24. Abowd, J.M. The U.S. Census Bureau Adopts Differential Privacy. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 18–23 August 2018; p. 2867.

25. Kasiviswanathan, S.P.; Lee, H.K.; Nissim, K.; Raskhodnikova, S.; Smith, A. What can we learn privately? *SIAM J. Comput.* **2011**, *40*, 793–826. [CrossRef]

26. Learning with Privacy at Scale. Available online: https://machinelearning.apple.com/research/learning-with-privacy-at-scale (accessed on 31 Dec 2017).

27. Erlingsson, Ú.; Pihur, V.; Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 1054–1067.

28. Ding, B.; Kulkarni, J.; Yekhanin, S. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*; Long Beach, CA, USA, 4–9 December 2017; pp. 3571–3580.

29. Wang, N.; Xiao, X.; Yang, Y.; Zhao, J.; Hui, S.C.; Shin, H.; Shin, J.; Yu, G. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; pp. 638–649.

30. Qin, Z.; Yang, Y.; Yu, T.; Khalil, I.; Xiao, X.; Ren, K. Heavy hitter estimation over set-valued data with local differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 192–203.

31. Cormode, G.; Kulkarni, T.; Srivastava, D. Marginal release under local differential privacy. In Proceedings of the 2018 International Conference on Management of Data, Houston, TX, USA, 10-15 June 2018; pp. 131–146.

32. Wang, D.; Gaboardi, M.; Xu, J. Empirical risk minimization in non-interactive local differential privacy revisited. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2018; pp. 965–974.

33. Wang, Y.; Tong, Y.; Shi, D. Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework. In Proceedings of the AAAI Conference on Artificial Intelligence, New York City, NY, USA, 7–12 February 2020; pp. 6283-6290.

34. Arachchige, P.C.M.; Bertok, P.; Khalil, I.; Liu, D.; Camtepe, S.; Atiquzzaman, M. Local Differential Privacy for Deep Learning. *IEEE Internet Things J.* **2019**, *7*, 5827–5842. [CrossRef]

35. Wang, T.; Blocki, J.; Li, N.; Jha, S. Locally differentially private protocols for frequency estimation. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Baltimore, MD, USA, 15–17 August 2017; pp. 729–745.

36. Ye, Q.; Hu, H. Local Differential Privacy: Tools, Challenges, and Opportunities. In Proceedings of the Workshop of Web Information Systems Engineering (WISE), Hong Kong, SAR, China, 26–30 November 2019; pp. 13–23.

37. Bebensee, B. Local differential privacy: A tutorial. *arXiv* **2019**, arXiv:1907.11908.

38. Li, N.; Ye, Q. Mobile Data Collection and Analysis with Local Differential Privacy. In Proceedings of the IEEE International Conference on Mobile Data Management (MDM), Hong Kong, SAR, China, 10–13 June 2019; pp. 4–7.

39. Zhao, P.; Zhang, G.; Wan, S.; Liu, G.; Umer, T. A survey of local differential privacy for securing internet of vehicles. *J. Supercomput.* **2019**, *76*, 1–22. [CrossRef]

40. Yang, M.; Lyu, L.; Zhao, J.; Zhu, T.; Lam, K.Y. Local Differential Privacy and Its Applications: A Comprehensive Survey. *arXiv* **2020**, arXiv:2008.03686.

41. Xiong, X.; Liu, S.; Li, D.; Cai, Z.; Niu, X. A Comprehensive Survey on Local Differential Privacy. *Secur. Commun. Netw.* **2020**, *2020*, 8829523. [CrossRef]

42. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Local privacy and statistical minimax rates. In Proceedings of the IEEE Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 429–438.

43. Warner, S.L. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.* **1965**, *60*, 63–69. [CrossRef] [PubMed]

44. Wang, Y.; Wu, X.; Hu, D. Using Randomized Response for Differential Privacy Preserving Data Collection. In Proceedings of the EDBT/ICDT Workshops, Bordeaux, France, 15-16 Mar 2016; Volume 1558, pp. 1–8.

45. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Proceedings of the Theory of Cryptography Conference, New York, NY, USA, 4–7 March 2006; pp. 265–284.

46. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS), Providence, RI, USA, 20–23 October 2007; Volume 7, pp. 94–103.

47. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Local privacy, data processing inequalities, and statistical minimax rates. *arXiv* **2013**, arXiv:1302.3203.

48. Joseph, M.; Mao, J.; Neel, S.; Roth, A. The Role of Interactivity in Local Differential Privacy. In Proceedings of the IEEE Annual Symposium on Foundations of Computer Science (FOCS), Baltimore, MD, USA, 9–12 November 2019; pp. 94–105.

49. Wang, D.; Xu, J. On sparse linear regression in the local differential privacy model. In Proceedings of the International Conference on Machine Learning, Long Beach, California, USA, 9–15 June 2019; pp. 6628–6637.

50. Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; Naor, M. Our data, ourselves: Privacy via distributed noise generation. In Proceedings of the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, 28 May–1 June 2006; pp. 486–503.

51. Bassily, R. Linear queries estimation with local differential privacy. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019; pp. 721–729.

52. Avent, B.; Korolova, A.; Zeber, D.; Hovden, T.; Livshits, B. BLENDER: Enabling local search with a hybrid differential privacy model. In Proceedings of the USENIX Security Symposium, Vancouver, BC, Canada, 16–18 August 2017; pp. 747–764.

53. Andrés, M.; Bordenabe, N.; Chatzikokolakis, K.; Palamidessi, C. Geo-Indistinguishability: Differential Privacy for Location-Based Systems. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 901–914.

54. Alvim, M.S.; Chatzikokolakis, K.; Palamidessi, C.; Pazii, A. Metric-based local differential privacy for statistical applications. *arXiv* **2018**, arXiv:1805.01456.

55. Chatzikokolakis, K.; Andrés, M.E.; Bordenabe, N.E.; Palamidessi, C. Broadening the scope of differential privacy using metrics. In Proceedings of the International Symposium on Privacy Enhancing Technologies Symposium, Bloomington, IN, USA, 10–12 July 2013; pp. 82–102.

56. Gursoy, M.E.; Tamersoy, A.; Truex, S.; Wei, W.; Liu, L. Secure and Utility-Aware Data Collection with Condensed Local Differential Privacy. *IEEE Trans. Dependable Secur. Comput.* **2019**, 1–13. [CrossRef]

57. NIE, Y.; Yang, W.; Huang, L.; Xie, X.; Zhao, Z.; Wang, S. A Utility-Optimized Framework for Personalized Private Histogram Estimation. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 655–669. [CrossRef]

58. Murakami, T.; Kawamoto, Y. Utility-optimized local differential privacy mechanisms for distribution estimation. In Proceedings of the USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; pp. 1877–1894.

59. Gu, X.; Li, M.; Xiong, L.; Cao, Y. Providing Input-Discriminative Protection for Local Differential Privacy. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; pp. 505–516.

60. Takagi, S.; Cao, Y.; Yoshikawa, M. POSTER: Data Collection via Local Differential Privacy with Secret Parameters. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, 5–9 October 2020; pp. 910–912.

61. Bassily, R.; Smith, A. Local, private, efficient protocols for succinct histograms. In Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, Portland, OR, USA, 14–17 June 2015; pp. 127–135.

62. Wang, T.; Zhao, J.; Yang, X.; Ren, X. Locally differentially private data collection and analysis. *arXiv* **2019**, arXiv:1906.01777.

63. Kairouz, P.; Oh, S.; Viswanath, P. Extremal mechanisms for local differential privacy. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Quebec, Canada, 8–13 December 2014; pp. 2879–2887.

64. Kairouz, P.; Bonawitz, K.; Ramage, D. Discrete distribution estimation under local privacy. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 2436–2444.

65. Wang, S.; Huang, L.; Wang, P.; Deng, H.; Xu, H.; Yang, W. In *Private Weighted Histogram Aggregation in Crowdsourcing*; Springer, Germany, 2016; pp. 250–261.

66. Bloom, B.H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* **1970**, *13*, 422–426. [CrossRef]

67. Acharya, J.; Sun, Z.; Zhang, H. Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019; pp. 1120–1129.

68. Cormode, G.; Kulkarni, T.; Srivastava, D. Answering range queries under local differential privacy. *Proc. VLDB Endow.* **2019**, *12*, 1126–1138. [CrossRef]

69. Wang, S.; Huang, L.; Wang, P.; Nie, Y.; Xu, H.; Yang, W.; Li, X.Y.; Qiao, C. Mutual information optimally local private discrete distribution estimation. *arXiv* **2016**, arXiv:1607.08025.

70. Wang, S.; Huang, L.; Nie, Y.; Zhang, X.; Wang, P.; Xu, H.; Yang, W. Local Differential Private Data Aggregation for Discrete Distribution Estimation. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 2046–2059. [CrossRef]

71. Ren, X.; Yu, C.M.; Yu, W.; Yang, S.; Yang, X.; McCann, J. High-dimensional crowdsourced data distribution estimation with local privacy. In Proceedings of the IEEE International Conference on Computer and Information Technology (CIT), Nadi, Fiji, 8–10 December 2016; pp. 226–233.

72. Ren, X.; Yu, C.M.; Yu, W.; Yang, S.; Yang, X.; McCann, J.A.; Philip, S.Y. LoPub: High-Dimensional Crowdsourced Data Publication with Local Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2151–2166. [CrossRef]

73. Yang, X.; Wang, T.; Ren, X.; Yu, W. Copula-Based Multi-Dimensional Crowdsourced Data Synthesis and Release with Local Privacy. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Singapore, 4–8 December 2017; pp. 1–6.

74. Zhang, Z.; Wang, T.; Li, N.; He, S.; Chen, J. 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, Toronto, ON, Canada, 15–19 October 2018; pp. 212–229.

75. Wang, T.; Li, N.; Jha, S. Locally differentially private heavy hitter identification. *IEEE Trans. Dependable Secure Comput.* **2019**, 1–12. [CrossRef]

76. Wang, S.; Huang, L.; Nie, Y.; Wang, P.; Xu, H.; Yang, W. PrivSet: Set-Valued Data Analyses with Locale Differential Privacy. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 1088–1096.

77. Zhao, X.; Li, Y.; Yuan, Y.; Bi, X.; Wang, G. LDPart: Effective Location-Record Data Publication via Local Differential Privacy. *IEEE Access* **2019**, *7*, 31435–31445. [CrossRef]

78. Mishra, N.; Sandler, M. Privacy via pseudorandom sketches. In Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Chicago, IL, USA, 26–28 June 2006; pp. 143–152.

79. Hsu, J.; Khanna, S.; Roth, A. Distributed private heavy hitters. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Warwick, UK, 9–13 July 2012; pp. 461–472.

80. Bassily, R.; Nissim, K.; Stemmer, U.; Thakurta, A.G. Practical locally private heavy hitters. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 2288–2296.

81. Bun, M.; Nelson, J.; Stemmer, U. Heavy hitters and the structure of local privacy. In Proceedings of the ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, 10–15 June 2018; pp. 435–447.

82. Jia, J.; Gong, N.Z. Calibrate: Frequency Estimation and Heavy Hitter Identification with Local Differential Privacy via Incorporating Prior Knowledge. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2008–2016.

83. Sun, C.; Fu, Y.; Zhou, J.; Gao, H. Personalized privacy-preserving frequent itemset mining using randomized response. *Sci. World J.* **2014**, *2014*, 686151. [CrossRef]

84. Wang, T.; Li, N.; Jha, S. Locally differentially private frequent itemset mining. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–23 May 2018; pp. 127–143.

85. Thakurta, A.G.; Vyrros, A.H.; Vaishampayan, U.S.; Kapoor, G.; Freudinger, J.; Prakash, V.V.; Legendre, A.; Duplinsky, S. Emoji Frequency Detection and Deep Link Frequency. U.S. Patent 9,705,908, 19 June 2017.

86. Tang, J.; Korolova, A.; Bai, X.; Wang, X.; Wang, X. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv* **2017**, arXiv:1709.02753.

87. Fanti, G.; Pihur, V.; Erlingsson, Ú. Building a with the unknown: Privacy-preserving learning of associations and data dictionaries. *Priv. Enhancing Technol.* **2016**, *2016*, 41–61. [CrossRef]

88. Wang, N.; Xiao, X.; Yang, Y.; Hoang, T.D.; Shin, H.; Shin, J.; Yu, G. PrivTrie: Effective frequent term discovery under local differential privacy. In Proceedings of the IEEE ICDE, Paris, France, 16–19 April 2018; pp. 821–832.

89. Kim, S.; Shin, H.; Baek, C.; Kim, S.; Shin, J. Learning New Words from Keystroke Data with Local Differential Privacy. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 479–491. [CrossRef]

90. Ye, Q.; Hu, H.; Meng, X.; Zheng, H. PrivKV: Key-Value Data Collection with Local Differential Privacy. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019.

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

92. Sun, L.; Zhao, J.; Ye, X.; Feng, S.; Wang, T.; Bai, T. Conditional Analysis for Key-Value Data with Local Differential Privacy. *arXiv* **2019**, arXiv:1907.05014.

93. Gu, X.; Li, M.; Cheng, Y.; Xiong, L.; Cao, Y. PCKV: Locally Differentially Private Correlated Key-Value Data Collection with Optimized Utility. *arXiv* **2019**, arXiv:1911.12834.

94. Wang, S.; Nie, Y.; Wang, P.; Xu, H.; Yang, W.; Huang, L. Local private ordinal data distribution estimation. In Proceedings of the IEEE INFOCOM, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.

95. Yang, J.; Cheng, X.; Su, S.; Chen, R.; Ren, Q.; Liu, Y. Collecting Preference Rankings Under Local Differential Privacy. In Proceedings of the IEEE ICDE, Macau, China, 8–11 April 2019; pp. 1598–1601.

96. Black, D. *The Theory of Committees and Elections*; Springer Science & Business Media: Berlin, Germany, 2012.

97. Reilly, B. Social choice in the south seas: Electoral innovation and the borda count in the pacific island countries. *Int. Political Sci. Rev.* **2002**, *23*, 355–372. [CrossRef]

98. Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; Procaccia, A.D. *Handbook of Computational Social Choice*; Cambridge University Press: Cambridge, UK, 2016.

99. Wang, S.; Du, J.; Yang, W.; Diao, X.; Liu, Z.; Nie, Y.; Huang, L.; Xu, H. Aggregating Votes with Local Differential Privacy: Usefulness, Soundness vs. Indistinguishability. *arXiv* **2019**, arXiv:1908.04920.

100. Li, C.; Hay, M.; Miklau, G.; Wang, Y. A Data- and Workload-Aware Query Answering Algorithm for Range Queries Under Differential Privacy. *Proc. VLDB Endow.* **2014**, *7*, 341–352. [CrossRef]

101. Alnemari, A.; Romanowski, C.J.; Raj, R.K. An Adaptive Differential Privacy Algorithm for Range Queries over Healthcare Data. In Proceedings of the IEEE International Conference on Healthcare Informatics, Park City, UT, USA, 23–26 August 2017; pp. 397–402.

102. Kulkarni, T. Answering Range Queries Under Local Differential Privacy. In Proceedings of the 2019 International Conference on Management of Data, Amsterdam, The Netherlands, 30 June–5 July 2019; pp. 1832–1834.

103. Gu, X.; Li, M.; Cao, Y.; Xiong, L. Supporting Both Range Queries and Frequency Estimation with Local Differential Privacy. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 124–132.

104. Li, Z.; Wang, T.; Lopuhaä-Zwakenberg, M.; Li, N.; Škoric, B. Estimating Numerical Distributions under Local Differential Privacy. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 621–635.

105. Wang, T.; Yang, X.; Ren, X.; Yu, W.; Yang, S. Locally Private High-dimensional Crowdsourced Data Release based on Copula Functions. *IEEE Trans. Serv. Comput.* **2019**, 1–14. [CrossRef]

106. Qardaji, W.; Yang, W.; Li, N. PriView: Practical differentially private release of marginal contingency tables. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, Snowbird, UT, USA, 22–27 June 2014; pp. 1435–1446.

107. Wang, T.; Ding, B.; Zhou, J.; Hong, C.; Huang, Z.; Li, N.; Jha, S. Answering multi-dimensional analytical queries under local differential privacy. In Proceedings of the International Conference Management of Data, Amsterdam, The Netherlands, 30 June–5 July 2019; pp. 159–176.

108. Xu, M.; Wang, T.; Ding, B.; Zhou, J.; Hong, C.; Huang, Z. DPSAaS: Multi-Dimensional Data Sharing and Analytics as Services under Local Differential Privacy. *Proc. VLDB Endow.* **2019**, *12*, 1862–1865. [CrossRef]

109. Xue, Q.; Zhu, Y.; Wang, J. Joint Distribution Estimation and Naïve Bayes Classification under Local Differential Privacy. *IEEE Trans. Emerg. Topics Comput.* **2019**, 1–11. [CrossRef]

110. Cao, Y.; Yoshikawa, M.; Xiao, Y.; Xiong, L. Quantifying Differential Privacy under Temporal Correlations. In Proceedings of the IEEE ICDE, San Diego, CA, USA, 19–22 Apr 2017; pp. 821–832.

111. Wang, H.; Xu, Z. CTS-DP: Publishing correlated time-series data via differential privacy. *Knowl. Based Syst.* **2017**, *122*, 167–179. [CrossRef]

112. Cao, Y.; Yoshikawa, M.; Xiao, Y.; Xiong, L. Quantifying Differential Privacy in Continuous Data Release Under Temporal Correlations. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 1281–1295. [CrossRef] [PubMed]

113. Joseph, M.; Roth, A.; Ullman, J.; Waggoner, B. Local Differential Privacy for Evolving Data. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, 3–8 December 2018; pp. 2381–2390.

114. Dwork, C.; Naor, M.; Pitassi, T.; Rothblum, G.N. Differential privacy under continual observation. In Proceedings of the ACM Symposium on Theory of Computing, Cambridge, Massachusetts, USA, 5–8 June 2010; pp. 715–724.

115. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Minimax optimal procedures for locally private estimation. *J. Am. Stat. Assoc.* **2018**, *113*, 182–201. [CrossRef]

116. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership inference attacks against machine learning models. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 3–18.

117. Song, C.; Ristenpart, T.; Shmatikov, V. Machine learning models that remember too much. In Proceedings of the ACM SIGSAC CCS, Dallas, TX, USA, 30 Oct–3 Nov 2017; pp. 587–601.

118. Fredrikson, M.; Jha, S.; Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the ACM SIGSAC CCS, Denver, CO, USA, 12–16 October 2015; pp. 1322–1333.

119. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the ACM SIGSAC CCS, Vienna, Austria, 24–28 October 2016; pp. 308–318.

120. Phan, N.; Wu, X.; Hu, H.; Dou, D. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In Proceedings of the IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; pp. 385–394.

121. Lee, J.; Kifer, D. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1656–1665.

122. Zhao, J.; Chen, Y.; Zhang, W. Differential Privacy Preservation in Deep Learning: Challenges, Opportunities and Solutions. *IEEE Access* **2019**, *7*, 48901–48911. [CrossRef]

123. Jayaraman, B.; Evans, D. Evaluating Differentially Private Machine Learning in Practice. In Proceedings of the USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; pp. 1895–1912.

124. Yilmaz, E.; Al-Rubaie, M.; Chang, J.M. Locally differentially private naive bayes classification. *arXiv* **2019**, arXiv:1905.01039.

125. Berrett, T.; Butucea, C. Classification under local differential privacy. *arXiv* **2019**, arXiv:1912.04629.

126. Kung, S.Y. *Kernel Methods and Machine Learning*; Cambridge University Press: Cambridge, UK, 2014.

127. Nissim, K.; Stemmer, U.; Vadhan, S. Locating a small cluster privately. In Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, San Francisco, CA, USA, 26 June–1 July 2016; pp. 413–427.

128. Su, D.; Cao, J.; Li, N.; Bertino, E.; Jin, H. Differentially Private K-Means Clustering. In Proceedings of the ACM Conf. Data and Application Security and Privacy, New Orleans, LA, USA, 9–11 March 2016; pp. 26–37.

129. Feldman, D.; Xiang, C.; Zhu, R.; Rus, D. Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. In Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Pittsburgh, PA, USA, 18–21 April 2017; pp. 3–16.

130. Nissim, K.; Stemmer, U. Clustering Algorithms for the Centralized and Local Models. In *Algorithmic Learning Theory*; Springer Verlag, Germany, 2018; pp. 619–653.

131. Sun, L.; Zhao, J.; Ye, X. Distributed Clustering in the Anonymized Space with Local Differential Privacy. *arXiv* **2019**, arXiv:1906.11441.

132. Karapiperis, D.; Gkoulalas-Divanis, A.; Verykios, V.S. Distance-aware encoding of numerical values for privacy-preserving record linkage. In Proceedings of the IEEE ICDE, San Diego, CA, USA, 19–22 April 2017; pp. 135–138.

133. Karapiperis, D.; Gkoulalas-Divanis, A.; Verykios, V.S. FEDERAL: A framework for distance-aware privacy-preserving record linkage. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 292–304. [CrossRef]

134. Li, Y.; Liu, S.; Wang, J.; Liu, M. A local-clustering-based personalized differential privacy framework for user-based collaborative filtering. In Proceedings of the International Conference on Database Systems for Advanced Applications, Suzhou, China, 27–30 March 2017; pp. 543–558.

135. Akter, M.; Hashem, T. Computing aggregates over numeric data with personalized local differential privacy. In Proceedings of the Australasian Conference on Information Sanition Security and Privacy, Auckland, New Zealand, 3–5 July 2017; pp. 249–260.

136. Xia, C.; Hua, J.; Tong, W.; Zhong, S. Distributed K-Means clustering guaranteeing local differential privacy. *Comput. Secur.* **2020**, *90*, 1–11. [CrossRef]

137. Chaudhuri, K.; Monteleoni, C.; Sarwate, A.D. Differentially private empirical risk minimization. *J. Mach. Learn. Res.* **2011**, *12*, 1069–1109.

138. Smith, A.; Thakurta, A.; Upadhyay, J. Is interaction necessary for distributed private learning? In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 58–77.

139. Zheng, K.; Mou, W.; Wang, L. Collect at once, use effectively: Making non-interactive locally private learning possible. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 4130–4139.

140. Wang, D.; Chen, C.; Xu, J. Differentially Private Empirical Risk Minimization with Non-convex Loss Functions. In Proceedings of the International Conference on Machine Learning, Long Beach, California, USA, 9–15 June 2019; pp. 6526–6535.

141. Wang, D.; Zhang, H.; Gaboardi, M.; Xu, J. Estimating Smooth GLM in Non-interactive Local Differential Privacy Model with Public Unlabeled Data. *arXiv* **2019**, arXiv:1910.00482.

142. Wang, D.; Xu, J. Principal component analysis in the local differential privacy model. *Theor. Comput. Sci.* **2020**, *809*, 296–312. [CrossRef]

143. Fan, W.; He, J.; Guo, M.; Li, P.; Han, Z.; Wang, R. Privacy preserving classification on local differential privacy in data centers. *J. Parallel Distrib. Comput.* **2020**, *135*, 70–82. [CrossRef]

144. Yin, C.; Zhou, B.; Yin, Z.; Wang, J. Local privacy protection classification based on human-centric computing. *Hum.-Centric Comput. Inf. Sci.* **2019**, *9*, 33. [CrossRef]

145. Van der Hoeven, D. User-Specified Local Differential Privacy in Unconstrained Adaptive Online Learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 14080–14089.

146. Jun, K.S.; Orabona, F. Parameter-Free Locally Differentially Private Stochastic Subgradient Descent. *arXiv* **2019**, arXiv:1911.09564.

147. Osia, S.A.; Shamsabadi, A.S.; Sajadmanesh, S.; Taheri, A.; Katevas, K.; Rabiee, H.R.; Lane, N.D.; Haddadi, H. A hybrid deep learning architecture for privacy-preserving mobile analytics. *IEEE Internet Things J.* **2020**, *7*, 4505–4518. [CrossRef]

148. Zhao, J. Distributed Deep Learning under Differential Privacy with the Teacher-Student Paradigm. In Proceedings of the Workshops of AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

149. Xu, C.; Ren, J.; She, L.; Zhang, Y.; Qin, Z.; Ren, K. EdgeSanitizer: Locally Differentially Private Deep Inference at the Edge for Mobile Data Analytics. *IEEE Internet Things J.* **2019**, *6*, 5140–5151. [CrossRef]

150. Pan, X.; Wang, W.; Zhang, X.; Li, B.; Yi, J.; Song, D. How you act tells a lot: Privacy-leaking attack on deep reinforcement learning. In Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems, Montreal, QC, Canada, 13–17 May 2019; pp. 368–376.

151. Gajane, P.; Urvoy, T.; Kaufmann, E. Corrupt bandits for preserving local privacy. In *Algorithmic Learning Theory*; Springer Verlag, Germany, 2018; pp. 387–412.

152. Basu, D.; Dimitrakakis, C.; Tossou, A. Differential Privacy for Multi-armed Bandits: What Is It and What Is Its Cost? *arXiv* **2019**, arXiv:1905.12298.

153. Ono, H.; Takahashi, T. Locally Private Distributed Reinforcement Learning. *arXiv* **2020**, arXiv:2001.11718.

154. Ren, W.; Zhou, X.; Liu, J.; Shroff, N.B. Multi-Armed Bandits with Local Differential Privacy. *arXiv* **2020**, arXiv:2007.03121.

155. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]

156. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]

157. Yang, Q.; Liu, Y.; Cheng, Y.; Kang, Y.; Chen, T.; Yu, H. Federated learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2019**, *13*, 1–207. [CrossRef]

158. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]

159. Li, Q.; Wen, Z.; He, B. Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *arXiv* **2019**, arXiv:1907.09693.

160. Zheng, H.; Hu, H.; Han, Z. Preserving User Privacy For Machine Learning: Local Differential Privacy or Federated Machine Learning. *IEEE Intell. Syst.* **2020**, 35, 5–14. [CrossRef]

161. Cao, H.; Liu, S.; Zhao, R.; Xiong, X. IFed: A novel federated learning framework for local differential privacy in Power Internet of Things. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1–13. [CrossRef]

162. Seif, M.; Tandon, R.; Li, M. Wireless federated learning with local differential privacy. *arXiv* **2020**, arXiv:2002.05151.

163. Geyer, R.C.; Klein, T.; Nabi, M. Differentially private federated learning: A client level perspective. *arXiv* **2017**, arXiv:1712.07557.

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

165. Truex, S.; Liu, L.; Chow, K.H.; Gursoy, M.E.; Wei, W. LDP-Fed: Federated learning with local differential privacy. In Proceedings of the ACM International Workshop on Edge Systems, Analytics and Networking, Heraklion, Greece, 27 Apr 2020; pp. 61–66.

166. Bhowmick, A.; Duchi, J.; Freudiger, J.; Kapoor, G.; Rogers, R. Protection against reconstruction and its applications in private federated learning. *arXiv* **2018**, arXiv:1812.00984.

167. Li, J.; Khodak, M.; Caldas, S.; Talwalkar, A. Differentially private meta-learning. *arXiv* **2019**, arXiv:1909.05830.

168. Liu, R.; Cao, Y.; Yoshikawa, M.; Chen, H. FedSel: Federated SGD under Local Differential Privacy with Top-k Dimension Selection. *arXiv* **2020**, arXiv:2003.10637.

169. Sun, L.; Qian, J.; Chen, X.; Yu, P.S. LDP-FL: Practical Private Aggregation in Federated Learning with Local Differential Privacy. *arXiv* **2020**, arXiv:2007.15789.

170. Naseri, M.; Hayes, J.; De Cristofaro, E. Toward Robustness and Privacy in Federated Learning: Experimenting with Local and Central Differential Privacy. *arXiv* **2020**, arXiv:2009.03561.

171. Apple iOS Security. Available online: https://developer.apple.com/documentation/security (accessed on 10 May 2019).

172. Kessler, S.; Hoff, J.; Freytag, J.C. SAP HANA goes private: From privacy research to privacy aware enterprise analytics. *Proc. VLDB Endow.* **2019**, *12*, 1998–2009. [CrossRef]

173. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. [CrossRef]

174. Usman, M.; Jan, M.A.; Puthal, D. PAAL: A Framework based on Authentication, Aggregation and Local Differential Privacy for Internet of Multimedia Things. *IEEE Internet Things J.* **2020**, *7*. [CrossRef]

175. Ou, L.; Qin, Z.; Liao, S.; Li, T.; Zhang, D. Singular Spectrum Analysis for Local Differential Privacy of Classifications in the Smart Grid. *IEEE Internet Things J.* **2020**, *7*, 5246–5255. [CrossRef]

176. Zhao, Y.; Zhao, J.; Yang, M.; Wang, T.; Wang, N.; Lyu, L.; Niyato, D.; Lam, K.Y. Local differential privacy based federated learning for Internet of Things. *arXiv* **2020**, arXiv:2004.08856.

177. Song, Z.; Li, Z.; Chen, X. Local Differential Privacy Preserving Mechanism for Multi-attribute Data in Mobile Crowdsensing with Edge Computing. In Proceedings of the IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; pp. 283–290.

178. Gaboardi, M.; Lim, H.W.; Rogers, R.M.; Vadhan, S.P. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 2111–2120.

179. Cai, B.; Daskalakis, C.; Kamath, G. Priv'IT: Private and sample efficient identity testing. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 635–644.

180. Sheffet, O. Differentially private ordinary least squares. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 Aug 2017; pp. 3105–3114.

181. Tong, X.; Xi, B.; Kantarcioglu, M.; Inan, A. Gaussian mixture models for classification and hypothesis tests under differential privacy. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy, Philadelphia, PA, USA, 19–21 July 2017; pp. 123–141.

182. Kairouz, P.; Oh, S.; Viswanath, P. Extremal Mechanisms for Local Differential Privacy. *J. Mach. Learn. Res.* **2016**, *17*, 1–51.

183. Sheffet, O. Locally Private Hypothesis Testing. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4612–4621.

184. Gaboardi, M.; Rogers, R. Local Private Hypothesis Testing: Chi-Square Tests. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1612–1621.

185. Gaboardi, M.; Rogers, R.; Sheffet, O. Locally Private Mean Estimation: Z-test and Tight Confidence Intervals. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019; pp. 2545–2554.

186. Acharya, J.; Canonne, C.; Freitag, C.; Tyagi, H. Test without Trust: Optimal Locally Private Distribution Testing. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Naha, Okinawa, Japan, 16–18 April 2019; pp. 2067–2076.

187. Qin, Z.; Yu, T.; Yang, Y.; Khalil, I.; Xiao, X.; Ren, K. Generating synthetic decentralized social graphs with local differential privacy. In Proceedings of the ACM SIGSAC CCS, Dallas, TX, USA, 30 October–3 November 2017; pp. 425–438.

188. Zhang, Y.; Wei, J.; Zhang, X.; Hu, X.; Liu, W. A two-phase algorithm for generating synthetic graph under local differential privacy. In Proceedings of the International Conference on Communication and Network Security, Beijing, China, 30 May–1 June 2018; pp. 84–89.

189. Liu, P.; Xu, Y.; Jiang, Q.; Tang, Y.; Guo, Y.; Wang, L.e.; Li, X. Local differential privacy for social network publishing. *Neurocomputing* **2020**, *391*, 273–279. [CrossRef]

190. Gao, T.; Li, F.; Chen, Y.; Zou, X. Preserving local differential privacy in online social networks. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, Guilin, China, 19–21 June 2017; pp. 393–405.

191. Gao, T.; Li, F.; Chen, Y.; Zou, X. Local differential privately anonymizing online social networks under hrg-based model. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 1009–1020. [CrossRef]

192. Yang, J.; Ma, X.; Bai, X.; Cui, L. Graph Publishing with Local Differential Privacy for Hierarchical Social Networks. In Proceedings of the IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 17–19 July 2020; pp. 123–126.

193. Ye, Q.; Hu, H.; Au, M.H.; Meng, X.; Xiao, X. Towards Locally Differentially Private Generic Graph Metric Estimation. In Proceedings of the IEEE ICDE, Dallas, TX, USA, 20–24 April 2020; pp. 1922–1925.

194. Sun, H.; Xiao, X.; Khalil, I.; Yang, Y.; Qin, Z.; Wang, H.; Yu, T. Analyzing subgraph statistics from extended local views with decentralized differential privacy. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 703–717.

195. Wei, C.; Ji, S.; Liu, C.; Chen, W.; Wang, T. AsgLDP: Collecting and Generating Decentralized Attributed Graphs With Local Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3239–3254. [CrossRef]

196. Jeong, H.; Park, B.; Park, M.; Kim, K.B.; Choi, K. Big data and rule-based recommendation system in Internet of Things. *Clust. Comput.* **2019**, *22*, 1837–1846. [CrossRef]

197. Calandrino, J.A.; Kilzer, A.; Narayanan, A.; Felten, E.W.; Shmatikov, V. You Might Also Like: Privacy Risks of Collaborative Filtering. In Proceedings of the IEEE Symposium on Security and Privacy (SP), Berkeley, California, USA, 22–25 May 2011; pp. 231–246.

198. Liu, R.; Cao, J.; Zhang, K.; Gao, W.; Liang, J.; Yang, L. When Privacy Meets Usability: Unobtrusive Privacy Permission Recommendation System for Mobile Apps Based on Crowdsourcing. *IEEE Trans. Serv. Comput.* **2018**, *11*, 864–878. [CrossRef]

199. Shin, H.; Kim, S.; Shin, J.; Xiao, X. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1770–1782. [CrossRef]

200. Jiang, J.Y.; Li, C.T.; Lin, S.D. Towards a more reliable privacy-preserving recommender system. *Inf. Sci.* **2019**, *482*, 248–265. [CrossRef]

201. Guo, T.; Luo, J.; Dong, K.; Yang, M. Locally differentially private item-based collaborative filtering. *Inf. Sci.* **2019**, *502*, 229–246. [CrossRef]

202. Gao, C.; Huang, C.; Lin, D.; Jin, D.; Li, Y. DPLCF: Differentially Private Local Collaborative Filtering. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 961–970.

203. Wang, S.; Huang, L.; Tian, M.; Yang, W.; Xu, H.; Guo, H. Personalized privacy-preserving data aggregation for histogram estimation. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.