electronics

MDPI

Article

Secure Medical Data Collection in the Internet of Medical Things Based on Local Differential Privacy

Jinpeng Wang and Xiaohui Li *

School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China
* Correspondence: dxxylxh@lnut.edu.cn

Abstract: As big data and data mining technology advance, research on the collection and analysis of medical data on the internet of medical things (IoMT) has gained increasing attention. Medical institutions often collect users' signs and symptoms from their devices for analysis. However, the process of data collection may pose a risk of privacy leakage without a trusted third party. To address this issue, we propose a medical data collection based on local differential privacy and Count Sketch (MDLDP). The algorithm first uses a random sampling technique to select only one symptom for perturbation by a single user. The perturbed data is then uploaded using Count Sketch. The third-party aggregates the user-submitted data to estimate the frequencies of the symptoms and the mean extent of their occurrence. This paper theoretically demonstrates that the designed algorithm satisfies local differential privacy and unbiased estimation. We also evaluated the algorithm experimentally with existing algorithms on a real medical dataset. The results show that the MDLDP algorithm has good utility for key-value type medical data collection statistics in the IoMT.

Keywords: local differential privacy; count sketch; medical data; internet of medical things

check for updates

Citation: Wang, J.; Li, X. Secure Medical Data Collection in the Internet of Medical Things Based on Local Differential Privacy. *Electronics* **2023**, *12*, 307. https://doi.org/10.3390/electronics12020307

Academic Editor: Andrei Kelarev

Received: 21 December 2022
Revised: 3 January 2023
Accepted: 5 January 2023
Published: 6 January 2023

1. Introduction

As technologies such as the internet of things, artificial intelligence, and big data continue to advance, digital medical technologies [1] are being applied more widely. With the widespread adoption of the internet of medical things (IoMT), users can easily upload relevant physical sign data and diagnostic results through their devices [2,3], including smartwatches, smartphones, and other wearable devices. This allows medical institutions to effectively monitor regional disease conditions by collecting and analyzing patient symptom data [4]. Additionally, the Centers for Disease Control and Prevention (CDC) can even predict further infectious disease outbreaks [5]. However, while people benefit from medical big data, user personal information and related data are highly vulnerable to leakage, which can be accessed and even maliciously used by others as the data is released and shared. The leakage of users' disease information can cause a serious psychological burden to users and negative social impacts [6]. To prevent this, the privacy protection of medical information is vital in the development of medical big data and the IoMT.

Health privacy is a major concern for individuals and society, as demonstrated by laws such as the European General Data Protection Regulation (GDPR) [7]. The IoMT involves the use of devices to collect and generate large amounts of health data, which can be vulnerable to privacy breaches from untrustworthy data collectors and attackers during the data collection and dissemination processes [8]. To protect the privacy of user data, various techniques have been developed, including identification [9], anonymity [10], and differential privacy [11,12]. The identification process adopts the identity authentication mechanism for each user, and the user must log in to the system with their own identity to ensure the security of the user's data. In the process of data collection, anonymity achieves the goal of privacy protection by deleting or hiding identifying attributes, and differential privacy protect data by adding noise. These techniques can provide a certain

level of privacy protection in specific scenarios, however, they may have limitations in their applicability to the IoMT and often require the involvement of trusted third parties for data collection. Local differential privacy (LDP) [13] is a privacy protection technique that aims to protect data privacy by adding noise to the data collection and analysis processes. It has been proposed as a solution that does not require a trusted third party and addresses the limitations of anonymity and differential privacy models. Research on how to implement local differential privacy protection is currently a focus in academia and industry.

In recent years, there has been a significant amount of research on local differential privacy, and it has been applied in various fields. Google uses the RAPPOR [14] method for data collection in the Chrome browser, Apple [15] uses local differential privacy algorithms for statistical analysis of user emoticons, and Samsung proposed Harmony [16], based on local differential privacy, for collecting and analyzing on-device data. However, there has not yet been a feasible privacy protection method proposed for key-value medical data under the IoMT, so this paper focuses on addressing this problem.

There are three main challenges in solving this problem. The first one is how to design a reasonable perturbation scheme for key-value type medical data that maintains the correlation between keys and values, thereby improving the accuracy when counting the data. The second one is to design data collection schemes that minimize noise error and communication costs for data collectors in the IoMT. The last is to ensure that the designed method should satisfy local differential privacy and unbiased estimation.

To address these challenges, we propose the MDLDP method, which is based on local differential privacy and the Count Sketch [17] structure. The main contributions of this paper are as follows:

1. To the first challenge, we propose MDLDP. Unlike existing algorithms that cannot handle key-valued data [18] or treat users' data keys and values separately, MDLDP perturbs data keys uniformly and locally, maintaining the correlation and availability between data.
2. To deal with the second challenge, we propose to encode and collect user data through the Count Sketch, which can be effectively used for large-scale data collection.
3. We theoretically analyzed that MDLDP satisfies ϵ-local differential privacy, and that frequency and mean estimations are unbiased. We also compared MDLDP in terms of mean square error and relative error for frequency and mean estimations through comparative experiments to verify its effectiveness in protecting the privacy of key-value type medical data in the IoMT.

2. Related Work

2.1. Privacy Protection in IoMT

The widespread adoption of IoMT devices in the healthcare industry has the potential to revolutionize patient care through the collection and analysis of real-time health data. However, the use of these devices also raises concerns about privacy and security, as the collected data may contain sensitive and personal information. To address these concerns, various privacy protection techniques have been proposed, including the use of secure communication protocols, anonymization techniques, and differential privacy [9–12].

While these techniques can provide some level of privacy protection, they have their own limitations and may not be fully applicable to the IoMT context. For example, secure communication protocols and anonymization techniques may not provide sufficient protection and may be unable to resist homogeneous attacks and background knowledge attacks. Differential privacy relies on trusted third parties, which can be vulnerable to data breaches. In this paper, we propose the use of local differential privacy [13] as a solution to address these limitations and protect the privacy of IoMT-collected healthcare data. Our proposed approach aims to solve the problem of user privacy protection in the IoMT by applying local differential privacy to the data collection process.

2.2. Local Differential Privacy

Unlike differential privacy, which relies on a trusted third party to add noise to the data, local differential privacy allows each user to locally perturb their own data using a random response mechanism [19] before submitting the noisy results to the untrusted data collector for privacy protection. Currently, the research on local differential privacy consists of frequency publishing, mean publishing, and frequent itemset mining. The RAPPOR [14] method, proposed by Google, uses Bloom filter [20] encoding and randomized response [19] to provide local protection for user browsing records. The SHist method [21], proposed by Cormode et al. uses hashing and matrix projection techniques to reduce communication costs. Wang et al. [22] propose the OUE and OLH methods, which use unary coding and local hashing to improve estimation accuracy. Ye et al. [23] propose the PrivKV method, a key-value type data collection scheme. Gu et al. [24] propose the PCKV algorithm, which optimizes the PrivKV method and uses a sampling-filling technique to collect key-value data without splitting the privacy budget.

In this paper, we proposed an algorithm that can effectively collect key-value type medical data in the IoMT. Our proposed methods address the limitations of existing approaches, which often struggle with the need for frequent segmentation of privacy budgets and the inability to handle sparse key-value domains.

3. Preliminaries and Problem Definition

3.1. Preliminaries

Definition 1. *ϵ-Local Differential Privacy. For a randomized algorithm $\mathcal{M}$ with the domain of definition $Dom(\mathcal{M})$ and domain of values $Range(\mathcal{M})$. It is possible for pairs of input values d and d' (where $d, d' \in Dom(\mathcal{M})$) to produce the same output $y \in Range(\mathcal{M})$, it holds:*

$$\Pr[\mathcal{M}(\mathrm{d}) = \mathrm{y}] \leq \mathrm{e}^{\epsilon} \times \Pr[\mathcal{M}(\mathrm{d}\prime) = \mathrm{y}], \tag{1}$$

The algorithm $\mathcal{M}$ satisfies ϵ-local differential privacy, where ϵ is the privacy budget. A smaller ϵ value indicates stronger privacy protection provided by $\mathcal{M}$.

Theorem 1. *Sequential Composition [25]. A set of randomized algorithms $\mathcal{M} = \left\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{\backslash}\right\}$ applied to the same dataset D. If $\mathcal{M}_{\backslash}(1 \leq i \leq n)$ satisfies ϵ-local differential privacy, $\mathcal{M}$ satisfies $(\sum_i \epsilon_i)$ local differential privacy on dataset D.*

Theorem 2. *Parallel Composition [25]. A dataset* D *that is divided into* n *disjoint subsets,* $\mathrm{D} = \{\mathrm{D1}, \mathrm{D2}, \ldots, \mathrm{Dn}\}$*, and an ϵ-local differential privacy algorithm $\mathcal{M}$. The algorithm $\mathcal{M}$ satisfies ϵ-local differential privacy when applied to each subset individually.*

Definition 2. *Randomized Response [19]. Let* v *be a user's binary value,* $\hat{\mathrm{v}}$ *is responded. For any* v*, there is:*

$$Pr[\hat{v} = v] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+1}, & \text{if } \hat{v} = v \\ \frac{1}{e^{\epsilon}+1}, & \text{if } \hat{v} \neq v \end{cases}, \tag{2}$$

The randomized response outputs the true value with probability $\frac{e^{\epsilon}}{e^{\epsilon}+1}$ and outputs the opposite value with probability $\frac{1}{e^{\epsilon}+1}$. However, the traditional randomized response technique is only suitable for binary attributes. To expand its applicability to a wider range of attributes, the generalized randomized response has been proposed.

Definition 3. *Generalized Randomized Response [26]. Let* $\mathrm{v} \in \mathrm{R}$ *be a user's input, where* R *is a set of the true values users can have and the length of* R *is* d, $\hat{\mathrm{v}}$ *is responded. The generalized randomized response works as:*

$$Pr[\hat{v} = v] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+d-1}, & \text{if } \hat{v} = v \\ \frac{1}{e^{\epsilon}+d-1}, & \text{if } \hat{v} \neq v \end{cases}, \tag{3}$$

The generalized randomized response is a statistical technique that is similar to the traditional randomized response technique, however, it is designed to be applicable to a wider range of attributes beyond just binary attributes.

Definition 4. *Count Sketch [17]. Count sketch is a data structure that is used to estimate the frequency of items in a large dataset. It consists of an array of counters and a hash function. The array of counters is used to store the frequency estimates for each item, and the hash function is used to map each item to a specific counter in the array. When an item is added to the count sketch, it is hashed using the hash function and the corresponding counter in the array is incremented. To estimate the frequency of an item, the item is hashed using the same hash function and the value of the corresponding counter is returned as an estimate of the frequency.*

Count Sketch allows for efficient frequency estimation of items in large datasets without the need to store the entire dataset in the memory and it is a useful tool for data analysis and has applications in a variety of fields, including machine learning, data mining, and information retrieval.

3.2. Problem Definition

This paper focuses on the problems of frequency estimation and mean estimation of medical data in the IoMT. Frequency estimates are used to measure symptoms, signs, etc., while mean estimates are used to measure their extent. In the following section, we will model these problems using local differential privacy.

Let U be a set of n users, $U = \{u_1, u_2, \ldots, u_n\}$. Let K be a set of d vital signs and symptoms, $K = \{k_1, k_2, \ldots, k_d\}$. The values of the original symptoms and vital signs are mapped to the interval $[0, 1]$ by scaling, and represented as V. For each user u_i, there is a set of symptoms $S_i = \{\langle k_j, v_j \rangle | 1 \leq j \leq l_i, k_j \in K, v_j \in V\}$ of length l_i, where each symptom k_j is associated with a corresponding value $v_j \in V$. For example, as shown in Figure 1, a user may have symptoms such as cough, fever, and sneezing, each with a corresponding degree of severity.

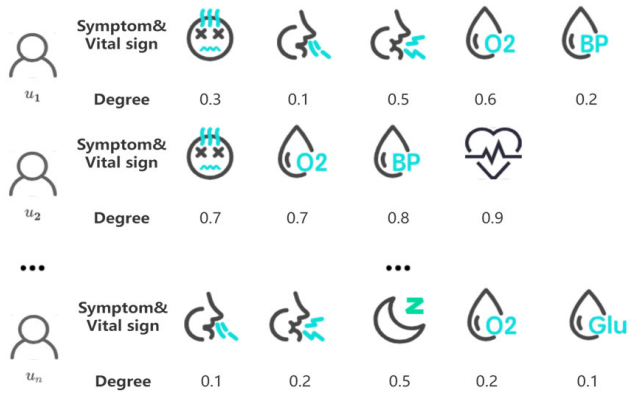


Figure 1. Symptoms and corresponding degree.

Here, we define the frequency and mean estimates of physical symptoms as follows:

Definition 5. *Frequency Estimation of Symptoms. The frequency estimate of a physical symptom is the number of u_i who have reported experiencing the symptom k and the value of symptom k is v in the set of u_i's all symptoms S_i, divided by the total number of n users in the sample. It is calculated as follows:*

$$f_k = \frac{|\{\sum_1^n u_i | \exists(k, \mathrm{v}) \in S_i\}|}{\mathrm{n}}, \tag{4}$$

Definition 6. *Mean Estimation of Symptoms. The mean estimate of a physical symptom is the average severity of the symptom among all users who have reported experiencing it. It is calculated by summing the values of the symptom for all users who have reported experiencing it, and dividing by the total number of users who have reported the symptom. It is calculated as follows:*

$$\mu_k = \frac{\{\sum v | (u_i \in U, \exists(k, v) \in S_i)\}}{n f_k}, \tag{5}$$

In practical usage scenarios, it is not necessary to have accurate estimates of vital signs and physical symptoms. For example, epidemic prevention authorities may only need to determine the general spread of a disease in a region based on specific symptoms, rather than precise counts. However, it is important to ensure the relative precision and availability of the data. Therefore, in this paper, we will use the mean square error (MSE) and relative error (RE) to measure the accuracy of frequency estimation and mean estimation and aim to obtain results with errors as small as possible.

4. Algorithm for Medical Data Collection under LDP

4.1. System Architecture

The system architecture used by MDLDP is shown in Figure 2, which shows the specific operation and interaction methods between users and third parties. The system architecture layout is relatively simple, reduces the system overhead, and has practicability and security.

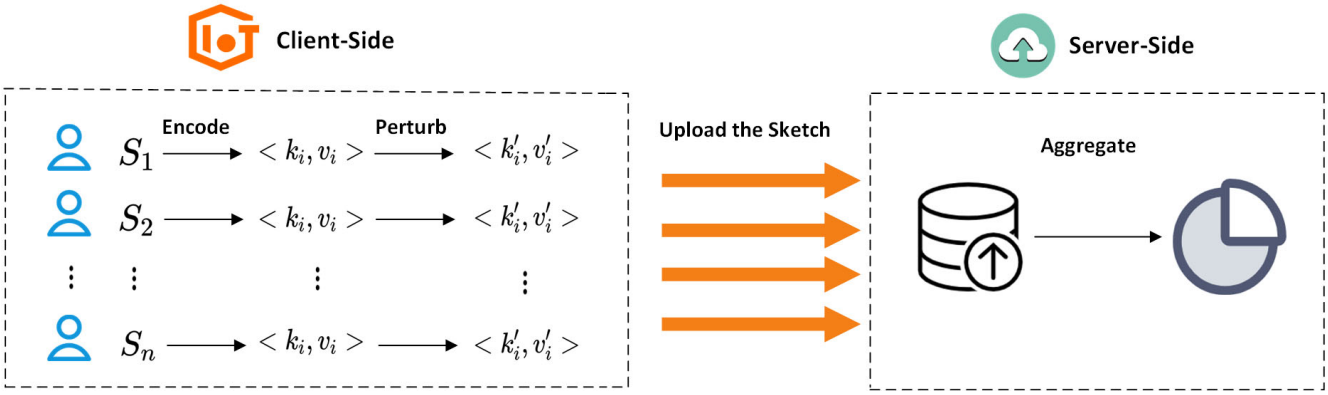


Figure 2. System architecture figure.

Users locally encode and perturb their sign and symptom degree. In the process of local encoding, the user data is extracted by sampling-filling technology. Then the local disturbance is carried out by GRR, and the disturbance result is uploaded. A single user only needs to upload one set of data. Then a third party aggregates the perturbed data for statistics. In the process of communication, Count Sketch is fully used to reduce the cost of user including computation, communication and storage.

4.2. Algorithm Design

The algorithm works by first encoding the original user data. Then, it randomly selects a data sample for discretization and perturbation, and maps the perturbated results

through Count Sketch. Finally, the data collector aggregates and reduces the statistics of the perturbed Sketch submitted by the user in order to obtain statistical information about the user data while protecting the privacy of the user. The overall design of MDLDP is illustrated in Figure 3.

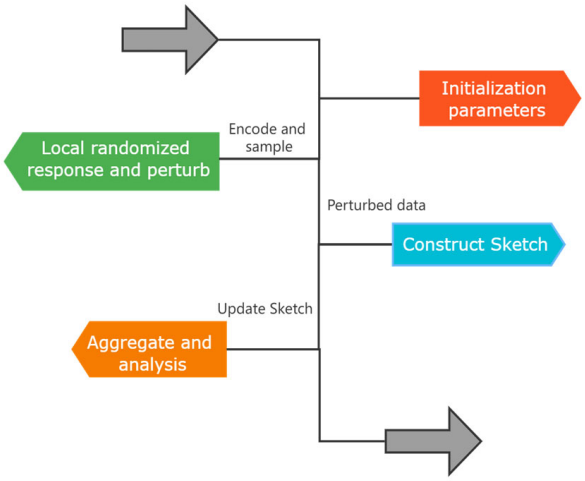


Figure 3. Workflow diagram of the MDLDP algorithm.

The MDLDP algorithm is based on local differential privacy and uses Count Sketch to collect large-scale data statistics. The specific implementation is shown in Algorithm 1.

Algorithm 1: MDLDP algorithm

Input: A set of key-value pairs $S = S_1, S_2, \ldots, S_n$ for n users, the value domain of symptoms $K = k_1, k_2, \ldots, k_d$, the size of symptoms d, perturbation probability p and q, error parameter ξ, probability confidence δ, privacy budget ϵ

Output: frequency estimation $\hat{f}$, mean estimation $\hat{\mu}$

1. $\hat{f} \leftarrow \varnothing$, $\hat{\mu} \leftarrow \varnothing$
2. $t \leftarrow \left\lceil \ln \frac{1}{\delta} \right\rceil$, $w \leftarrow \left\lceil \frac{1}{\xi^2} \right\rceil$
3. choose a set of 2-universal hash functions $H = h_1, h_2, \ldots, h_t$ to mapping K to $[w]$
4. choose a set of 2-universal hash functions $G = g_1, g_2, \ldots, g_t$ to mapping K to $\{-1, +1\}$
5. for $i \leftarrow 1$ to n do:
6. $\left(j, \left\langle k_j^*, v_j^* \right\rangle\right) \leftarrow$ MDLDP-LRR (S_i, ϵ)
7. $\mathcal{V}_i \leftarrow$ MDLDP-CS $\left(j, \left\langle k_j^*, v_j^* \right\rangle\right), H, G, t, w)$
8. end for
9. $\mathcal{V}, \mathcal{V}_-, \mathcal{V}_+ \leftarrow$ MDLDP-Aggregate $(\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n, t, w)$
10. for $j \leftarrow 1$ to d do:
11. $f\left(k_j\right), c_+(j), c_-(j) \leftarrow median\{\mathcal{V}[t][h_i(j)g_i(j))]|i \in [t]\}$
12. $\hat{f}\left(k_j\right) \leftarrow \frac{\mathrm{d} \times \hat{f}\left(\mathrm{k_j}\right) - (1-\mathrm{p})}{\mathrm{p}-\mathrm{q}}$
13. $\mathrm{c}_{\pm}^*(\mathrm{j}) \leftarrow \frac{\mathrm{c}_{\pm}(\mathrm{j}) + \mathrm{c}(\mathrm{j})(\mathrm{p}-1)}{2\mathrm{p}-1}$
14. $\hat{\mu}\left(k_j\right) \leftarrow \frac{c_+^*(j) - c_-^*(j)}{c_+^*(j) + c_-^*(j)}$
15. $\hat{f} \leftarrow \hat{f}\left(k_j\right) \cup \hat{f}, \hat{\mu} \leftarrow \hat{\mu}\left(k_j\right) \cup \hat{\mu}$,
16. end for
17. return $\hat{f}, \hat{\mu}$

The algorithm consists of four key steps: initialization, encoding and perturbation, sketch construction, and aggregation and data statistics. First, in the initialization process (lines 1–4), we set a preset hash functions H to map symptoms K to an $\frac{1}{\xi^2}$-length bit string and set hash functions G to map input value into $-1, +1$. Then, each user applies the MDLDP-LRR algorithm to perturb their original data (line 6). The perturbed data is then sent to a third party using the MDLDP-CS algorithm (line 7). Finally, the third party aggregates the data submitted by the users using the MDLDP-Aggregate algorithm (line 9), and computes the corrected statistical results (lines 10–16).

4.3. Local Randomized Response and Perturb

In this process, the user side will encode and perturb the operation using the MDLDP-LRR algorithm. Firstly, the set of key-value symptoms owned by the user is uniformly encoded (line 1). The symptoms owned by the user are encoded as $\langle 1, v_i \rangle$, and the symptoms not owned by the user are filled with $\langle 0, 0 \rangle$ to generate S'_i. After encoding, a random $\langle k_j, v_j \rangle$ of the j symptom is sampled (line 2) for local perturbation (lines 3–8). As shown in Algorithm 2.

Algorithm 2: MDLDP-LRR algorithm

Input: A key-value pair S_i for u_i, privacy budget ϵ

Output: Key number j, key-value $\left\langle k_j^*, v_j^* \right\rangle$

1. $S'_i \leftarrow$ Encoding (S_i)
2. $j \leftarrow random.randint\left(1, \text{len}\left(S'_i\right)\right)$
3. if the j-th $\left\langle k_j, v_j \right\rangle$ is $\langle 0, 0 \rangle$:
4. discretize and perturb $<k_j, v_j>$ as follows:

$$\left\langle k_j^*, v_j^* \right\rangle = \begin{cases} \langle 0, 0 \rangle, & w.p.p \\ \langle 1, 1 \rangle, & w.p.\frac{1-p}{2} \\ \langle 1, -1 \rangle, & w.p.\frac{1-p}{2} \end{cases}$$

5. else if the j-th $<k_j, v_j>$ is $\langle 1, v \rangle$:
6. discretize and perturb $<k_j, v_j>$ as follows:

$$\left\langle k_j^*, v_j^* \right\rangle = \begin{cases} \langle 0, 0 \rangle, & w.p.\frac{1+v}{2} \times \frac{1-p_1}{2} + \frac{1-v}{2} \times \frac{1-p_2}{2} \\ \langle 1, 1 \rangle, & w.p.\frac{1+v}{2} \times p_1 + \frac{1-v}{2} \times \frac{1-p_2}{2} \\ \langle 1, -1 \rangle, & w.p.\frac{1+v}{2} \times \frac{1-p_1}{2} + \frac{1-v}{2} \times p_2 \end{cases}$$

7. end if
8. return $\left(j, \left\langle k_j^*, v_j^* \right\rangle\right)$

Referring to papers [24,27,28], we designed the key-value perturbation schemes that do not require splitting privacy budgets. During the local perturbation process, the key value will undergo a discretization operation, and then a random response will be generated using the GRR algorithm to ensure ϵ-local differential privacy. The specific perturbation process is illustrated in Figure 4.

According to the result of the disturbance, the symptoms of user key data will be sent in three forms: $\langle 0, 0 \rangle, \langle 1, 1 \rangle, \langle 1, -1 \rangle$. These forms include p, p_1, and p_2, and maintain the original value probability after disturbance as Definition 3. Specifically, $p = e^\epsilon / (e^\epsilon + d - 1)$ where $d = 3$ to maintain the original value probability in the disturbance process. This means $p = e^\epsilon / (e^\epsilon + 2)$, $p_1 = p_2 = p$, and $q = (1 - p)/2$.

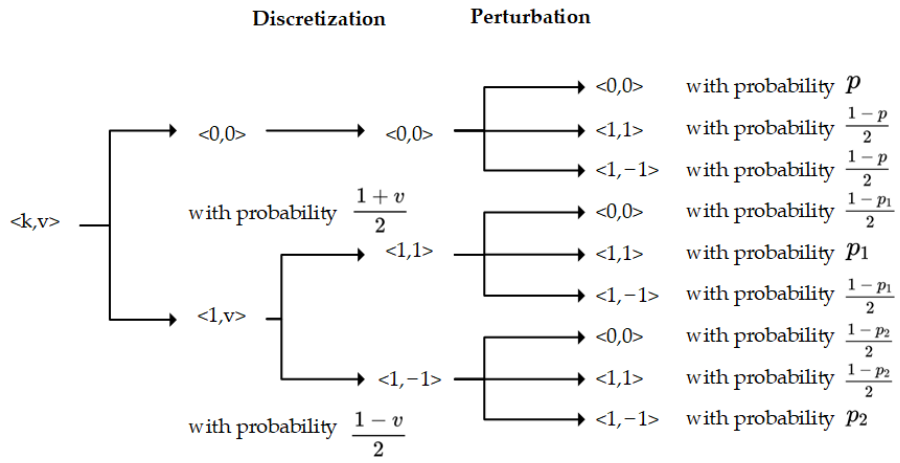


Figure 4. Local perturbation of MDLDP-LRR.

4.4. Construct Sketch

After the perturbation is completed, the user uploads the data through the MDLDP-CS algorithm. As shown in Algorithm 3.

Algorithm 3: MDLDP-CS algorithm

Input: The u_i's key-value pair $\left(j, \left\langle k_j^*, v_j^* \right\rangle\right)$ after perturbation, hash functions $H = h_1, h_2, \ldots, h_t$ and $G = g_1, g_2, \ldots, g_t$, the number of hash functions t, the bit number w
Output: matrix $\mathcal{V}_i$

1. $\mathcal{V}_\rangle \leftarrow memset(array, sizeof(array[t][w]))$
2. for $i \leftarrow 1$ to t do:
3. $\left(\mathcal{V}_\rangle[t][h_i(j)] = g_i(j) \times v_j^*\right.$
4. end for
5. return $\mathcal{V}_\rangle$

After locally perturbing the key-value data, the user maps the values into a hash matrix using the MDLDP-CS algorithm (lines 2–4) and then sends the resulting matrix to the data collector.

4.5. Aggregate and Analysis

After receiving the perturbed data from the user, the third party aggregates the sketch using the MDLDP-Aggregate method. The median value of symptom k is then chosen as the appropriate result (line 4) and further statistical analysis is conducted based on this. As shown in Algorithm 4.

The different values corresponding to the keys are aggregated into their respective matrices (lines 2–11). In Algorithm 1, the median method is used to select the appropriate result as the statistical value of symptom k (line 11), and further statistical analysis is conducted based on this result.

Algorithm 4: MDLDP-Aggregate algorithm

Input: All users' matrix $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n$, the number of hash functions t, the bit number w
Output: The aggregated $\mathcal{V}, \mathcal{V}_+, \mathcal{V}_-$

$\mathcal{V}, \mathcal{V}_+, \mathcal{V}_- \leftarrow memset(array, sizeof(array[t][w]))$
for $i \leftarrow 1$ to t do:
  for $j \leftarrow 1$ to w do:
$\mathcal{V}[i][j] = \mathcal{V}[i][j] + |\mathcal{V}_i[j]|$
    if $\mathcal{V}_i[j] = -1$ do:
$\mathcal{V}_-[i][j] = \mathcal{V}_-[i][j] + \mathcal{V}_i[j]$
    else:
$\mathcal{V}_+[i][j] = \mathcal{V}_+[i][j] + \mathcal{V}_i[j]$
    end if
  end for
end for
return $\mathcal{V}, \mathcal{V}_+, \mathcal{V}_-$

4.6. Algorithm Analysis

In the following, the MDLDP algorithm will be analyzed and verified from two aspects of privacy and practicality.

4.6.1. Privacy Analysis

Theorem 3. *MDLDP satisfies ϵ-local differential privacy.*

Proof of Theorem 3. In the MDLDP-LRR algorithm, the user only needs to spend their privacy budget ϵ on local perturbation. The consumption of the privacy budget is not involved in the other steps of the algorithm. Consider two different key-value pairs $\langle k_1, v_1 \rangle$ and $\langle k_2, v_2 \rangle$, and the resulting perturbed key-value pair $\langle k^*, v^* \rangle$. As shown in Figure 4, the result of the perturbation for the key-value pairs $\langle k_1, v_1 \rangle$ and $\langle k_2, v_2 \rangle$ is $\langle 0, 0 \rangle$, $\langle 1, 1 \rangle$, and $\langle 1, -1 \rangle$, which can be divided into the following three cases to demonstrate that the algorithm satisfies the ϵ-local differential privacy requirement.

1. $< k^*, v^* \geq \langle 0, 0 \rangle$:

$$\begin{aligned} \frac{\Pr[perturb(\langle k_1, v_1 \rangle) = \langle 0,0 \rangle]}{\Pr[perturb(\langle k_2, v_2 \rangle) = \langle 0,0 \rangle]} &\leq \frac{\Pr[perturb(\langle 0,0 \rangle) = \langle 0,0 \rangle]}{\Pr[perturb(\langle 1, \pm 1 \rangle) = \langle 0,0 \rangle]} \\ &= \frac{p}{\frac{1+v}{2} \times \frac{1-p_1}{2} + \frac{1-v}{2} \times \frac{1-p_2}{2}} = e^{\epsilon} \end{aligned} \tag{6}$$

2. $< k^*, v^* \geq \langle 1, 1 \rangle$:

$$\begin{aligned} \frac{\Pr[perturb(\langle k_1, v_1 \rangle) = \langle 1,1 \rangle]}{\Pr[perturb(\langle k_2, v_2 \rangle) = \langle 1,1 \rangle]} &\leq \frac{\Pr[perturb(\langle 1,1 \rangle) = \langle 1,1 \rangle]}{\Pr[perturb(\langle 0,0 \rangle) = \langle 1,1 \rangle]} \\ &= \frac{p_1}{\frac{1-p}{2}} = e^{\epsilon} \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{\Pr[perturb(\langle k_1, v_1 \rangle) = \langle 1,-1 \rangle]}{\Pr[perturb(\langle k_2, v_2 \rangle) = \langle 1,-1 \rangle]} &\leq \frac{\Pr[perturb(\langle 1,-1 \rangle) = \langle 1,-1 \rangle]}{\Pr[perturb(\langle 1,1 \rangle) = \langle 1,-1 \rangle]} \\ &= \frac{p_2}{\frac{1-p_1}{2}} = e^{\epsilon} \end{aligned} \tag{8}$$

3. $< k^*, v^* \geq \langle 1, -1 \rangle$:

$$\begin{aligned} \frac{\Pr[perturb(\langle k_1, v_1 \rangle) = \langle 1, -1 \rangle]}{\Pr[perturb(\langle k_2, v_2 \rangle) = \langle 1, -1 \rangle]} &\leq \frac{\Pr[perturb(\langle 1, -1 \rangle) = \langle 1, -1 \rangle]}{\Pr[perturb(\langle 0, 0 \rangle) = \langle 1, -1 \rangle]} \\ &= \frac{p_2}{\frac{1-p_1}{2}} = e^{\epsilon} \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{\Pr[perturb(\langle k_1, v_1 \rangle) = \langle 1, -1 \rangle]}{\Pr[perturb(\langle k_2, v_2 \rangle) = \langle 1, -1 \rangle]} &\leq \frac{\Pr[perturb(\langle 1, -1 \rangle) = \langle 1, -1 \rangle]}{\Pr[perturb(\langle 1, 1 \rangle) = \langle 1, -1 \rangle]} \\ &= \frac{p_2}{\frac{1-p_1}{2}} = e^{\epsilon} \end{aligned} \tag{10}$$

According to the above three cases, it can be deduced that the perturbation of data in MDLDP partially satisfies ϵ-local differential privacy.

4.6.2. Practical Analysis

Cormode et al. in Fact 3.5 [29] proved that the Count Sketch returns an unbiased estimator for any point query. So, we have just proven that the MDLDP algorithm satisfies the unbiased estimate with the reduction statistics.

Theorem 4. *MDLDP satisfies the unbiased estimate. In MDLDP, under the perturbation with the probability* $p = \frac{e^{\epsilon}}{e^{\epsilon}+2}$ *and* $q = \frac{1}{e^{\epsilon}+2}$, *both the frequency estimate* $\hat{f}$ *of the key and the mean estimate* $\hat{\mu}$ *of the corresponding value of the key satisfy the unbiased estimate.*

Proof of Theorem 4. We prove from two aspects.

1. $E\left[\hat{f}\right] = f$

User u_i obtains the j-th key-value pair by sampling, and each key is sampled with the probability $f_j = \frac{1}{d}$. Sampling n users, the number of j-th key is equal to 1 is $c_{k=1}$, and its probability is $f_{k=1} = \frac{c_{k=1}}{n}$. Disturbance sampling keys for the probability of 0 and 1, respectively, $Pr[k = 0] = (1 - f_{k=1}) \times p + f_{k=1} \times q$ and $Pr[k = 1] = (1 - f_{k=1}) \times (1 - p) + f_{k=1} \times (1 - q)$. The sampling satisfies a binomial distribution, to construct the likelihood function $L(f_{k=1}) = Pr[k = 1]^{c_{k=1}} \times (Pr[k = 0])^{\frac{n}{d} - c_{k=1}}$. The likelihood function is calculated by taking the ln of both sides of L and taking the derivative of the L to deduce the estimator $\hat{f}_k = \frac{d \times f_k - (1-p)}{p-q}$, so there is:

$$\begin{aligned} E\left(\hat{f}_k\right) &= E\left[\frac{d \times f_k - (1-p)}{p-q}\right] = \frac{d \times E[f_k] - (1-p)}{p-q} \\ &= \frac{f_k \times (1-q) + (1-f_k) \times (1-p) - (1-p)}{p-q} \\ &= f_k \end{aligned} \tag{11}$$

2. $E[\hat{\mu}] = \mu$

Assuming v_j represents the real value and v_j^* represents the disturbed value, Formula (5) demonstrates that to prove $E[\hat{\mu}_k] = \mu_k$, just prove $E\left[v_j^*\right] = v_j$. So, there is:

$$\begin{aligned} E\left(v_j^*\right) &= \sum_1^n v_j^* \times \Pr\left(v_j^*\right) \\ &= 0 \times \left[\frac{1}{2}p + \frac{1}{2}\left((1+v_j)p_1 + (1-v_j)\frac{1-p_2}{2}\right)\right] \\ &+1 \times \left[\frac{1}{2}\left(\frac{1-p}{2}\right) + \frac{1}{2}\left(\frac{1+v_j}{2}p_1 + \frac{1-v_j}{2}\frac{1-p_2}{2}\right)\right] \\ &+(-1) \times \left[\frac{1}{2}\left(\frac{1-p}{2}\right) + \frac{1}{2}\left(\frac{1-v_j}{2}p_2 + \frac{1+v_j}{2}\frac{1-p_1}{2}\right)\right] \\ &= \frac{v_j(e^{\epsilon}-1)}{2(e^{\epsilon}+2)} \end{aligned} \tag{12}$$

In order to satisfy $E\left[v_j^*\right] = v_j$, it is necessary to set the correction factor $\frac{2(e^\epsilon+2)}{e^\epsilon-1}$. So that $E\left[v_j^*\right] \times \frac{2(e^\epsilon+2)}{e^\epsilon-1} = v_j$ which leads to $E[\hat{\mu}_k] = \mu_k$.

4.6.3. Complexity Analysis

The MDLDP algorithm has a time complexity of $O\left(ln\frac{1}{\delta}\right)$ and a space complexity of $O\left(\frac{1}{\xi^2}ln\frac{1}{\delta}\right)$ in each phase. In the initialization phase, the time complexity is $O\left(\frac{1}{\xi^2}ln\frac{1}{\delta}\right)$, as the algorithm creates $\frac{1}{\xi^2} \times ln\frac{1}{\delta}$ hash functions and $\frac{1}{\xi^2} \times ln\frac{1}{\delta}$ counters. The space complexity is also $O\left(\frac{1}{\xi^2}ln\frac{1}{\delta}\right)$, as the MDLDP-CS algorithm uses $\frac{1}{\xi^2} \times ln\frac{1}{\delta}$ counters to store the values. When users update the data, the time complexity is $O(1)$, as the algorithm only updates a single counter for each item. The space complexity remains $O\left(\frac{1}{\xi^2}ln\frac{1}{\delta}\right)$. In the aggregation phase, the time complexity is $O\left(\frac{1}{\xi^2}ln\frac{1}{\delta}\right)$, as the algorithm must compute the values of all $\frac{1}{\xi^2}ln\frac{1}{\delta}$ hash functions for the given item and retrieve the corresponding counters. The space complexity is $O(1)$, as the algorithm only stores a single result.

5. Experiment Analysis

5.1. Experiment Environment and Data Set

We used a laptop with 16 GB RAM and an AMD Ryzen 7 6800 H processor, running Windows 11, to implement and test the algorithms in Python.

To evaluate our algorithm, we used the MIMIC-III [30] (Medical Information Mart for Intensive Care) database, a large open-source dataset of intensive care unit patient data. The experimental data for this paper consisted of the D_ITEMS and LABEVENTS tables, which contain a total of 12,488 items and 76,075 records.

The experiment consists of two parts: (1) evaluating the effect of different error ξ and confidence δ parameters on the performance of the MDLDP algorithm under the same privacy budget and (2) comparing the effect of the MDLDP algorithm with constant ξ error and δ confidence to other local differential privacy algorithms under different privacy budgets. The experimental parameters are shown in Table 1.

Table 1. Experimental parameters table.

Parameters	Default Value	Range of Values
Numbers of items	12,488	
Numbers of data	76,075	
Error	0.07	[0.01, 0.05, 0.07, 0.10]
Confidence probability	0.005	[0.005, 0.025, 0.10, 0.20]
Privacy budget	0.7	[0.1, 0.3, 0.5, 0.7]

The experiment will be assessed using the mean squared error (MSE) and relative error (RE) for frequency and mean estimation. Where f_k is the true frequency of the item key, $\hat{f}_k$ is the estimated frequency, μ is the true mean of the corresponding value, and $\hat{\mu}_k$ is the estimated mean. The relevant metrics are:

Definition 7. *Mean Squared Error (MSE). MSE is a commonly used error measurement method used to quantify the difference between predicted and actual values. It is defined as follows:*

$$MSE_{freq} = \frac{1}{d} \times \sum_{k \in K} \left(f_k - \hat{f}_k\right)^2 \tag{13}$$

Definition 8. *Relative Error (RE). RE is a commonly used error measurement method used to quantify the difference between predicted and actual values. It is defined as follows:*

$$\mathrm{RE} = \mathrm{Stat}_{k \in K} \frac{\left| f_k - \hat{f}_k \right|}{f_k} \tag{14}$$

The smaller MSE value and RE value indicate that the difference between predicted and actual values is smaller, and the model's accuracy is higher.

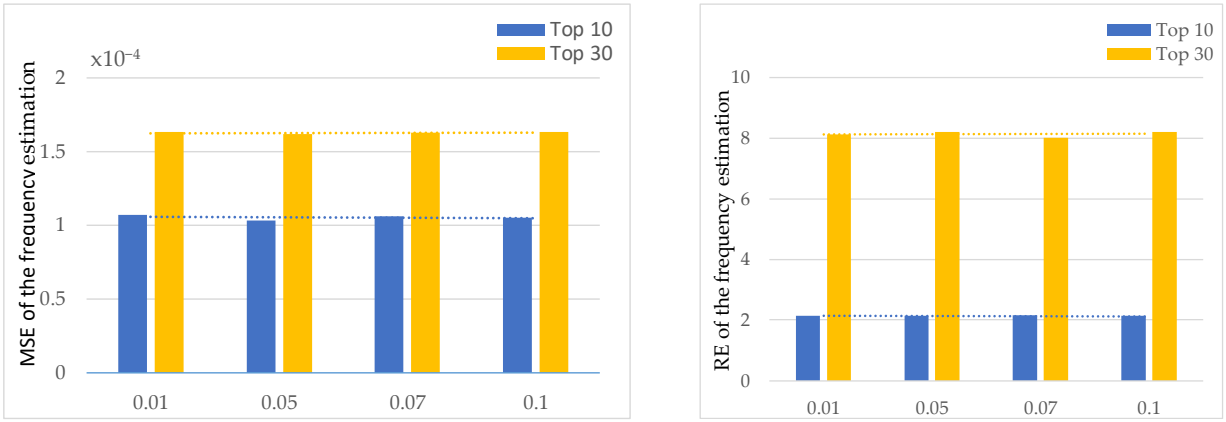
5.2. Experiment Comparison Analysis

5.2.1. Impact of Parameters

To investigate the influence of the error ξ, the confidence probability δ, and the privacy budget ϵ parameters on the MDLDP algorithm, we first specify these parameters. Then we confirm the protocol's validity by calculating the MSE and RE of the frequency estimation for the top 10 and top 30 most frequent elements. And we have counted the experimental run time under these parameters. We also repeat the experiment several times to reduce error.

First, we set confidence probability $\delta = 0.005$ and privacy budget $\epsilon = 0.7$. Error parameter ξ was $[0.01, 0.05, 0.07, 0.10]$ to test the effect of the error parameter.

Figure 5a,b show MSE and RE of the frequency estimation. The intuition is that ξ impacts less on MDLDP, due to the RE and MSE is not as ξ increased from 0.01 to 0.10.



(**a**) MSE with ξ (**b**) RE with ξ

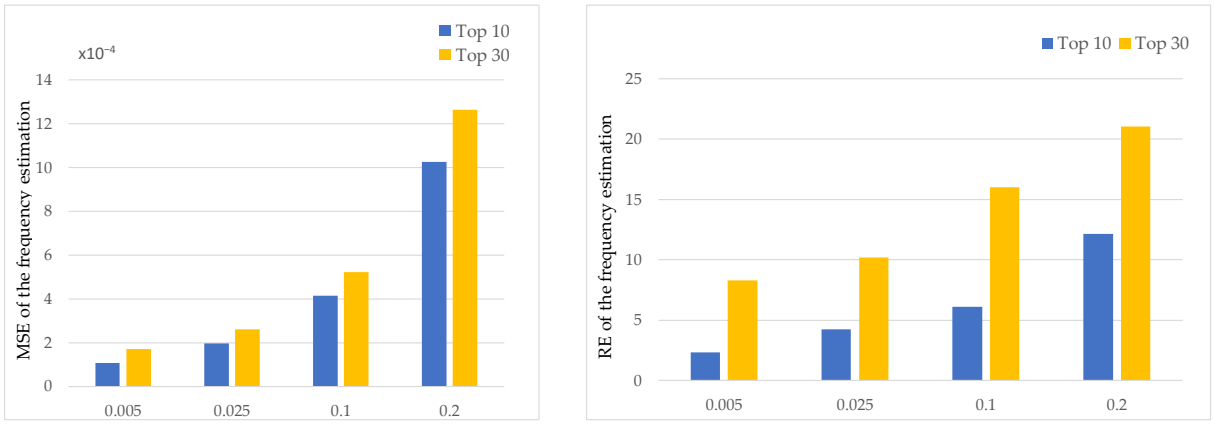
Figure 5. MSE and RE of the frequency estimation with different ξ.

Second, we set the confidence probability $\xi = 0.07$ and privacy budget $\epsilon = 0.7$. Confidence probability δ is $[0.005, 0.025, 0.10, 0.20]$ to test the effect of the confidence probability.

Figure 6a,b show the MSE and RE of frequency estimation. The intuition is that RE and MSE increased with the increase of δ. This means that increasing the number of hash functions can reduce the accuracy of the perturbed data.

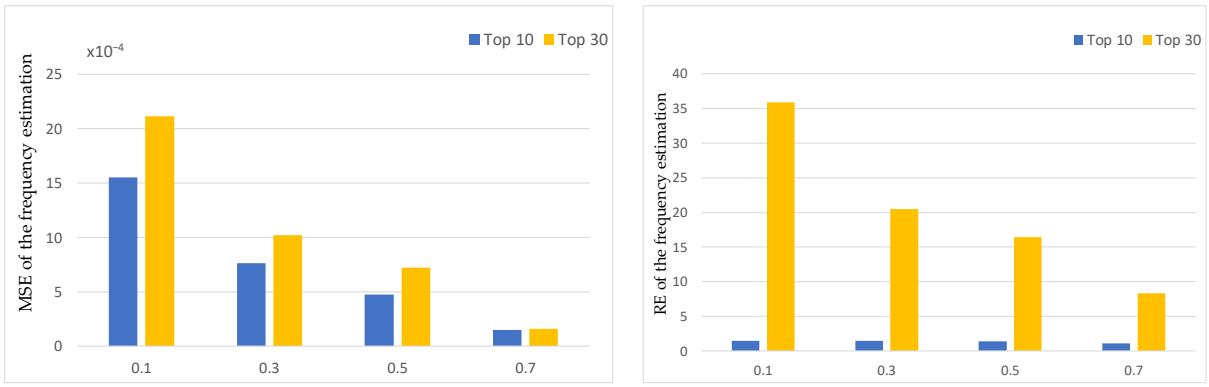
Third, we set the confidence probability $\xi = 0.07$ and confidence probability $\delta = 0.005$. The privacy budget ϵ was $[0.1, 0.3, 0.5, 0.7]$ to test the effect of privacy budget.

Figure 7a,b show the MSE and RE of frequency estimation. The intuition is that RE and MSE increased with the increase of privacy budget ϵ. This means more privacy budget can increase the accuracy.



(**a**) MSE with δ (**b**) RE with δ

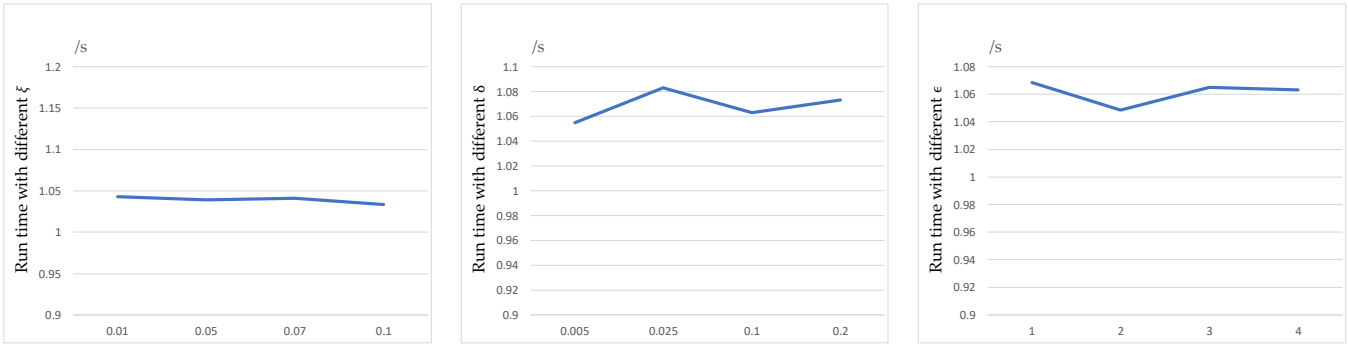
Figure 6. MSE and RE of the frequency estimation with different δ.



(**a**) MSE with ϵ (**b**) RE with ϵ

Figure 7. MSE and RE of the frequency estimation with different ϵ.

Finally, we counted the effect of the different parameters set in the above experiments on the algorithm running time. Based on Figure 8a–c, we can infer that these parameters have basically no effect on the algorithm running time with the current data set. As depicted in Figure 9, the running time of the MDLDP algorithm is linearly related to the size of the input data. The *X*-axis in the figure represents the size of the data randomly drawn from the dataset, while the *Y*-axis shows the corresponding running time of the algorithm. The results demonstrate that the algorithm is able to efficiently process large datasets and maintains good scalability as the volume of data increases.



(**a**) Run time with ξ (**b**) Run time with δ (**c**) Run time with ϵ

Figure 8. Run time with different parameters.

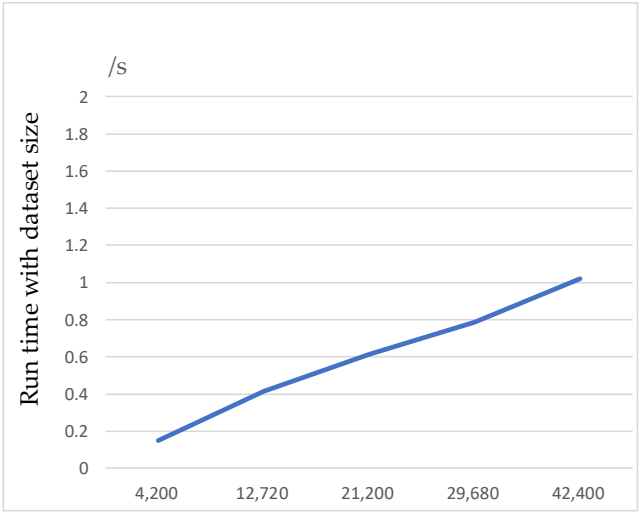


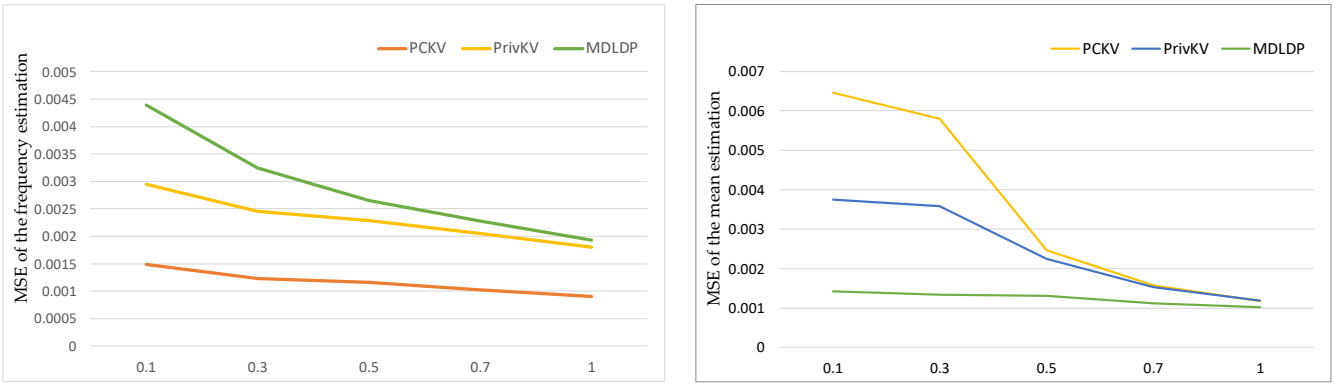
Figure 9. Run time with dataset size.

Through our experiments, we investigated the effect of parameter settings on the MDLDP algorithm. Firstly, we discovered that error parameter ξ had a limited effect on accuracy. Secondly, we observed that a higher confidence probability δ resulted in lower accuracy. Finally, we found that increasing privacy budget ϵ had a positive impact on the accuracy of frequency estimation.

5.2.2. Compare to Other Algorithms

We used MSE and RE to compare the MDLDP algorithm with error parameter $\xi = 0.007$ and confidence probability $\delta = 0.05$ to the PrivKV [23] and PCKV [24] algorithms, under different privacy budgets in $[0.1, 0.3, 0.5, 0.7]$.

In Figure 10a,b, we present the MSEs of frequency and mean estimates under different privacy budgets. The results indicate that the MDLDP algorithm performs better than the other algorithms, achieving lower MSE values under the same privacy budget. This demonstrates the effectiveness of the MDLDP algorithm in providing accurate frequency and mean estimates while preserving privacy.



(**a**) MSE of frequency estimation with ϵ (**b**) MSE of mean estimation with ϵ

Figure 10. MSE of the frequency estimation and mean estimation with different ϵ.

Figure 11a–d present the RE of the top 10–50 elements' frequency estimation for privacy budgets ϵ of $[0.1, 0.3, 0.5, 0.7]$. It can be seen from these figures that (1) the relative error of the statistics rises with the number of statistics; (2) the proposed algorithm has a higher accuracy under the same privacy budget.

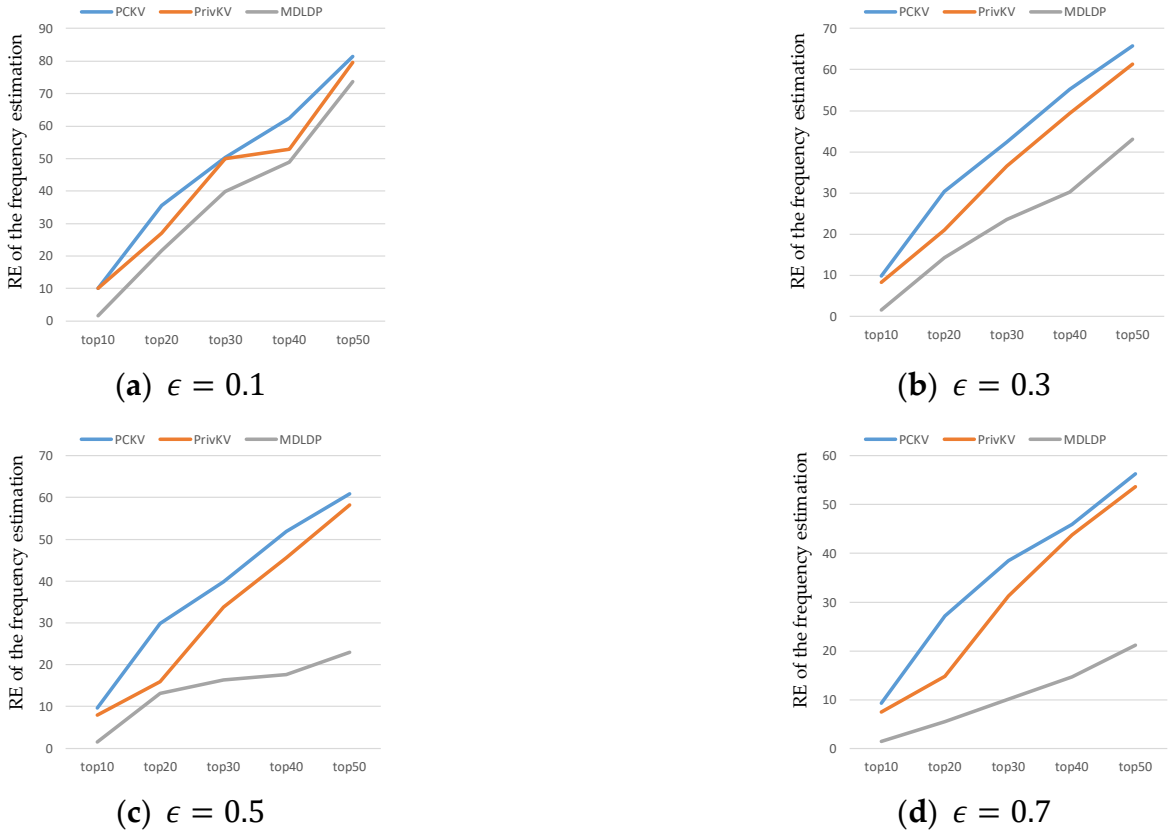


Figure 11. RE of the frequency estimation with different ϵ.

As shown in Figure 12, we compared the running times of the three algorithms under different privacy budgets ϵ of $[0.1, 0.3, 0.5, 0.7]$. It can indicate that the three algorithms have similar running times on the current experimental dataset, and the proposed algorithm performs better in frequency estimation and mean estimation with a similar run time to the other algorithms.

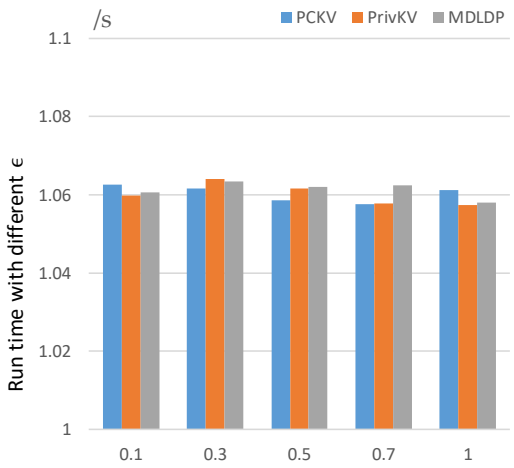


Figure 12. Run time with different ϵ.

The experimental comparison reveals that the MDLDP algorithm proposed has higher data availability than other local differential privacy protection algorithms. The results of our experiments suggest that the MDLDP algorithm can effectively protect medical data collection in the IoMT with high usability.

6. Conclusions

This paper focuses on local differential privacy algorithms for medical data collection in the IoMT and investigates how to estimate the frequency and mean of symptom occurrence under local differential privacy. We propose the MDLDP algorithm and design a random response perturbation that protects key-value type data while preserving the correlations between data, without splitting the privacy budget. We also designed a data collection method using Count Sketch based on local differential privacy. We demonstrate that the algorithm satisfies local differential privacy and unbiased estimation. Through experimental comparisons, we study the impact of different parameters on the performance of the MDLDP algorithm and show that it has higher usability and accuracy in the key-value type medical data collection process compared to other algorithms. In the future, we plan to design better encoding methods and optimize the perturbation technique to improve accuracy further.

Author Contributions: Conceptualization, J.W. and X.L.; methodology, J.W.; software, J.W.; validation, J.W.; formal analysis, J.W.; investigation, J.W.; resources, X.L.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, X.L.; visualization, J.W.; supervision, X.L.; project administration, X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Liaoning Applied Basic Research Program grant number 2022JH2/101300278.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sapci, A.H.; Sapci, H.A. Digital continuous healthcare and disruptive medical technologies: M-Health and telemedicine skills training for data-driven healthcare. *J. Telemed. Telecare* **2019**, *25*, 623–635. [CrossRef] [PubMed]
2. Vishnu, S.; Ramson SR, J.; Jegan, R. Internet of Medical Things (IoMT)—An Overview. In Proceedings of the 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 5–6 March 2020; pp. 101–104.
3. Keikhosrokiani, P. Predicating smartphone users' behaviour towards a location-aware IoMT-based information system: An empirical study. *Int. J. E-Adopt. IJEA* **2021**, *13*, 52–77. [CrossRef]
4. Singh, R.P.; Javaid, M.; Haleem, A.; Vaishya, R.; Ali, S. Internet of Medical Things (IoMT) for orthopaedic in COVID-19 pandemic: Roles, challenges, and applications. *J. Clin. Orthop. Trauma* **2020**, *11*, 713–717. [CrossRef] [PubMed]
5. Awotunde, J.B.; Folorunso, S.O.; Ajagbe, S.A.; Garg, J.; Ajamu, G.J. AiIoMT: IoMT-Based System-Enabled Artificial Intelligence for Enhanced Smart Healthcare Systems. In *Machine Learning for Critical Internet of Medical Things*; Springer: Cham, Switzerland, 2022; pp. 229–254.
6. Deep, S.; Zheng, X.; Jolfaei, A.; Yu, D.; Ostovari, P.; Bashir, A.K. A Survey of Security and Privacy Issues in the Internet of Things from the Layered Context. *Trans. Emerg. Telecommun. Technol.* **2020**, *33*, e3935. [CrossRef]
7. Voigt, P.; von dem Bussche, A. *The EU General Data Protection Regulation (Gdpr). A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10, pp. 10–381.
8. Garg, N.; Wazid, M.; Singh, J.; Singh, D.P.; Das, A.K. Security in IoMT-driven smart healthcare: A comprehensive review and open challenges. *Secur. Priv.* **2022**, *5*, e235. [CrossRef]
9. Wang, T.; Zheng, Z.; Bashir, A.K.; Jolfaei, A.; Xu, Y. FinPrivacy: A privacy-preserving mechanism for fingerprint identification. *ACM Trans. Internet Technol. TOIT* **2021**, *21*, 1–15. [CrossRef]
10. Weng, J.H.; Chi, P.W. Multi-Level Privacy Preserving K-Anonymity. In Proceedings of the 2021 16th Asia Joint Conference on Information Security (AsiaJCIS), Seoul, Republic of Korea, 19–20 August 2021; pp. 61–67.
11. Zhang, Z.; Wu, T.; Sun, X.; Yu, J. MPDP k-medoids: Multiple partition differential privacy preserving k-medoids clustering for data publishing in the Internet of Medical Things. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 15501477211042543. [CrossRef]
12. Lv, Z.; Piccialli, F. The security of medical data on internet based on differential privacy technology. *ACM Trans. Internet Technol.* **2021**, *21*, 1–18. [CrossRef]
13. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Local privacy and statistical minimax rates. In Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 429–438.
14. 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.

15. Team Apple Differential Privacy. *Learning with Privacy at Scale*; Apple: Cupertino, CA, USA, 2017; Volume 1, pp. 1–25.
16. 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.
17. Charikar, M.; Chen, K.; Farach-Colton, M. Finding Frequent Items in Data Streams. In *International Colloquium on Automata, Languages, and Programming, Proceedings of the International Colloquium on Automata, Languages, and Programming, Malaga, Spain, 8–13 July 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 693–703.
18. Wu, X.; Khosravi, M.R.; Qi, L.; Ji, G.; Dou, W.; Xu, X. Locally private frequency estimation of physical symptoms for infectious disease analysis in Internet of Medical Things. *Comput. Commun.* **2020**, *162*, 139–151. [CrossRef] [PubMed]
19. Warner, S.L. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.* **1965**, *60*, 63–69. [CrossRef] [PubMed]
20. Bruck, J.; Gao, J.; Jiang, A. Weighted Bloom Filter. In Proceedings of the 2006 IEEE International Symposium on Information Theory, Seattle, WA, USA, 9–14 July 2006; pp. 2304–2308.
21. Cormode, G.; Kulkarni, T.; Srivastava, D. Answering Range Queries under Local Differential Privacy. *Proc. VLDB Endow.* **2019**, *12*, 1126–1138. [CrossRef]
22. 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), Vancouver, BC, Canada, 16–18 August 2017; pp. 729–745.
23. Ye, Q.; Hu, H.; Meng, X.; Zheng, H. PrivKV: Key-Value Data Collection with Local Differential Privacy. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 317–331.
24. Gu, X.; Li, M.; Cheng, Y.; Xiong, L.; Cao, Y. {PCKV}: Locally Differentially Private Correlated {Key-Value} Data Collection with Optimized Utility. In Proceedings of the 29th USENIX Security Symposium (USENIX security 20), Boston, MA, USA, 12–14 August 2020; pp. 967–984.
25. McSherry, F.D. Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, Providence, RI, USA, 29 June–2 July 2009; pp. 19–30.
26. Christofides, T.C. A generalized randomized response technique. *Metrika* **2003**, *57*, 195–200. [CrossRef]
27. Zhang, X.; Fu, N.; Meng, X. Key-value data collection under local differential privacy. *Chin. J. Comput.* **2020**, *43*, 1479–1492. (In Chinese)
28. Zhang, X.; Xu, Y.; Fu, N.; Meng, X. Towards Private Key-Value Data Collection with Histogram. *J. Comput. Res. Dev.* **2021**, *58*, 624–637.
29. Cormode, G.; Yi, K. *Small Summaries for Big Data*; Cambridge University Press: Cambridge, UK, 2020.
30. Johnson, A.E.W.; Pollard, T.J.; Shen, L.; Lehman, L.H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L.A.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 160035. [CrossRef] [PubMed]