Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Effective truth discovery under local differential privacy by leveraging noise-aware probabilistic estimation and fusion[☆]

Pengfei Zhang, Xiang Cheng*, Sen Su, Ning Wang

State Key Laboratory of Networking and Switching Technology, No. 10, Xitucheng Road, Haidian District, 100876, Beijing, China
Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Haidian District, 100876, Beijing, China

ARTICLE INFO

Article history:
Received 13 July 2022
Received in revised form 30 November 2022
Accepted 15 December 2022
Available online 21 December 2022

Keywords:
Mobile crowdsensing
Truth discovery
Privacy protection
Local differential privacy

ABSTRACT

Truth discovery is an effective way to eliminate data inconsistency by integrating different worker-provided values. Although directly conducting non-private truth discovery approaches based on uploaded noisy values after adding Laplace noise for continuous inputs guarantees rigorous local differential privacy (LDP), it may result in poor performance due to the lot of contained noise. First, the injected noise for privacy protection randomly sampled from Laplace distribution may be excessive even with a large privacy budget, as the above distribution is unbounded and drops sharply with respect to the x-axis. Built-in Gaussian noise also usually exists within these uploaded noisy values, which may also have a negative effect on the aggregated truths under LDP and makes the problem investigated in this paper far more challenging. In this paper, we focus on obtaining accurate truths in the above cases under rigorous LDP for continuous inputs, and present a novel solution *TESLA*. The key idea of this solution is that we let injected noise for privacy protection and inherent Gaussian noise only weakly negatively affect the weight estimation and true aggregation. In particular, we design a runtime filtering mechanism (*RFM*) to obtain the supremum and infimum for the values after adding Laplace noise by considering these two types of noise together. Moreover, we develop a probabilistic fusion mechanism (*PFM*) to get the fused values by adaptively using the obtained supremum and infimum. Furthermore, we devise a probabilistic weight mechanism (*PWM*) to obtain a more accurate weight for each worker. Therefore, truth discovery can be conducted based on the new weight of each worker and the filtered values. We provide theoretical analyses of *TESLA*'s utility, privacy and complexity. Experimental results demonstrate the effectiveness and efficiency of *TESLA*. We also extend and verify *TESLA* over typical mean estimation as well as standard deviation calculation, and various machine learning tasks (e.g., logistic regression, support vector machine (*SVM*) and neural network). Experimental results also demonstrate its superiority.

1. Introduction

Crowdsensing systems have become an increasingly popular sensing paradigm that aggregate sensory data from a crowd of participating workers. Due to the diversity in sensing efforts, sensor qualities, ambient noises, etc., data inconsistency arises from multiple observed values drawn from different workers. For example, customer information can be found from multiple databases in a company, and erroneous customer account information in a company database may cause financial losses [1]. A patient's medical records may be scattered across different hospitals, and an incorrect diagnosis based on incorrect measurements of a patient may lead to serious consequences [2], and scientific discoveries may be guided in the wrong direction if they are derived from incorrect data [3].

Intuitively, we can compute the mean of all provided values, which would not be able to derive accurate aggregated results, because it regards all values equally. An ideal approach should have the capability to capture the difference in the quality of values among different participating workers. However, the challenge is that the reliability level (referred to as weight) of each worker is usually unknown as a prior. To address this challenge, truth discovery [4], is an effective technique that can jointly discover truthful information and estimate worker quality from noisy or even conflicting crowdsourced sensory data, which has

[☆] This work is the results of the research project funded by the National Natural Science Foundation of China [Grant No. 62072052 and No. 61872045] and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China [Grant No. 61921003].

* Corresponding author at: State Key Laboratory of Networking and Switching Technology, No. 10, Xitucheng Road, Haidian District, 100876, Beijing, China.

E-mail addresses: zpf.bupt@bupt.edu.cn (P. Zhang), chengxiang@bupt.edu.cn (X. Cheng), susen@bupt.edu.cn (S. Su), wangn@bupt.edu.cn (N. Wang).

https://doi.org/10.1016/j.knosys.2022.110213
0950-7051/© 2022 Elsevier B.V. All rights reserved.

attracted the interest of many researchers recently. It relies more on workers who contribute high-quality answers to derive the aggregated truth. Additionally, it usually consists of a truth aggregation phase and a weight estimation phase, and the server iteratively conducts these two phases until convergence.

Despite valuable truths that can potentially be obtained by truth discovery, uploading the raw values provided by each worker without proper privacy protection might put an individual's privacy in jeopardy. For example, customers may not want to share their account information to avoid potential financial losses, the medical records of a patient should be clearly protected, and the scientific discoveries from an individual should also be carefully protected. Participating workers are willing to contribute real data to crowdsensing systems only after privacy protection.

Local differential privacy (LDP) [5] has recently emerged as the de facto standard for individual privacy protection. It only collects randomized answers from each worker with a guarantee of plausible deniability, which can be implemented in a simple and efficient way through adopting the Laplace mechanism [6] for continuous inputs. Recently, the Laplace mechanism has been used by [6] to protect the continuous gradient information and numerical information.

In this paper, we aim to infer the truth for each task with high utility under rigorous LDP. Several studies have explored the problem of truth discovery under LDP, such as [7–12]. However, they are either designed specifically for discrete inputs or only satisfy the weaker versions of LDP for continuous inputs. Although directly conducting non-private truth discovery approaches after adopting the Laplace mechanism over continuous inputs guarantees rigorous LDP, poor performance may result due to a lot of contained noise. The reasons can be explained as follows. First of all, this process may inject a lot of noise based on the characteristics of the Laplace distribution. Specifically, the injected noise for privacy protection sampled from the Laplace distribution is random and unbounded, and this distribution decreases markedly sharply with respect to the x-axis, which is likely to produce extreme injected noise even with a large privacy budget ϵ. Moreover, existing studies [4,9] have shown that there usually exists built-in Gaussian noise within the worker-provided values, which has not been explored in sufficient detail and may also have a negative effect on the aggregated truths under rigorous LDP. It makes the problem investigated in this paper far more challenging. To our best knowledge, we are not aware of any other studies performed to date that can find the aggregated truths with high utility while satisfying rigorous ϵ-LDP.

To fill this gap, in this paper, we present a novel solution called *TESLA* (**T**ruth discov**E**ry via probabilistic e**S**timation mall under rigorous **L**ocal differential priv**A**cy) that can theoretically ensure privacy, utility and complexity, which is. The key idea of *TESLA* is to let injected noise for privacy protection and inherent Gaussian noise only weakly negatively affect weight estimation and true aggregation. In *TESLA*, we estimate the supremum and infimum for each worker-provided noisy value to bound the contained noise to be in favor of both weight estimation and true aggregation. We also estimate workers' weight distributions by jointly considering the above two types of noise to be beneficial to weight estimation.

To mitigate the negative influence of these two types of noise on *truth aggregation*, we first define a probability comparison function. Then, based on the defined probability comparison function, by leveraging the method of binary search, we design a runtime filtering mechanism (*RFM*) to obtain the supremum and infimum of the noisy values. Finally, given the supremum and infimum, we further develop a probabilistic fusion mechanism (*PFM*) to get the fused values. To mitigate negative influence of the above two kinds of noise on weight calculation, we design a probabilistic weight mechanism (*PWM*), which will serve as weight estimation of truth discovery. Specifically, we formulate a constrained nonlinear programming problem by modeling the mixed error distribution and derive an effective solution using the Lagrange multipliers approach.

Overall, *TESLA* works as follows. A worker first adds Laplace noise to his real values x_i and uploads the noisy value $\tilde{x}_i$ to the server. Then, the server invokes *RFM* and *PFM* successively to obtain the fused value $\hat{x}_i$. Finally, the server conducts truth discovery while adopting *CRH* (Conflict Resolution on Heterogeneous data) [13], which is one of the most representative approaches, to obtain the aggregated truth $\hat{x}^*$ for each task, where *PWM* serves as the weight estimation.

The key contributions are summarized as follows:

- We present a novel solution *TESLA* and give theoretical analysis of its utility, privacy and complexity. While being proposed for task truth discovery under rigorous LDP for continuous inputs, its idea is also applicable to other crowdsensing tasks (e.g., machine learning) while guaranteeing rigorous LDP.
- To mitigate the negative influence on truth aggregation, we first design *RFM* to obtain the supremum and infimum of the noisy values. Moreover, we develop *PFM* to obtain the fused values. Furthermore, we measure the performance of *PFM* in terms of (α, β)-accuracy.
- To mitigate the negative influence on weight calculation, we design *PWM*, which serves as weight estimation of the potential truth discovery approach.
- Extensive experimental results over two real-world datasets and two synthetic datasets demonstrate the effectiveness and efficiency of *TESLA*.
- We extend and verify *TESLA* for mean estimation and standard deviation calculation, which are two typical tasks for continuous inputs under LDP. Moreover, we also extend it for multiple machine learning tasks (e.g., logistic regression, support vector machine (*SVM*), neural network) over another synthetic dataset. Experimental results show that the extended solutions can also obtain good utility.

The rest of this paper is organized as follows. We discuss related work in Section 2. We describe the preliminaries in Section 3. The details of *TESLA* are presented in Section 4. The experimental results are discussed and analyzed in Section 5. Finally, we summarize our work in Section 6.

2. Related work

2.1. Truth discovery approaches

Many studies have explored the problem of truth discovery. Li et al. [13] propose the *CRH* (Conflict Resolution on Heterogeneous data) approach by iteratively conducting weight estimation and truth aggregation until convergence. To improve the utility of truth discovery for the long-tail data, they [4] devise a new weight estimation method through variance estimation. Xiao et al. [14] further extend the approach in [4]. Meng et al. [15] utilize the correlation between tasks to improve the utility of truth discovery. Wang et al. [16] focus on obtaining reliable truths from distributed data [16]. Ye et al. [17] adopt pattern discovery for truth discovery.

There also exist studies which focus on special scenarios, such as distributed data [16], text data [18] and time series data [19]. Wang et al. [16] focus on obtaining reliable truths from distributed data [16]. Zhang et al. [18] devise an approach to get reliable information from text data through truth discovery. Yao et al. [19] focus on online truth discovery for time series data.

Since we focus on ordinary numerical data in the mobile crowdsensing and *CRH* is the most representative approach, which has been widely used for the non-private comparison by the existing privacy-preserving truth discovery studies [7–9], we also adopt it for the non-private version in this paper.

2.2. Methods under Local Differential Privacy

Since the formulation of LDP [20], there are two lines along the study of LDP, which are the LDP mechanisms and LDP-based applications, such as mean estimation. The studies of LDP mechanisms focus on designing novel LDP methods which can serve as the basic building blocks for achieving LDP. The studies of LDP applications aim at adopting or improving the existing LDP mechanisms to collect certain data for the specific applications.

For LDP mechanisms, the popular randomized response [21] could be used for collecting binary attributes, which has been extended by Kairouz [22] to collect multiple attributes. Google [23] propose *RAPPOR* to collect users' homepage distributions. Wang et al. [24] propose an optimized local hashing method to mitigate the negative impact of large domains. These excellent studies can be used for collecting discrete data while we focus on continuous inputs. Thus, they cannot be applied by our setting.

For LDP applications, Duchi [20] propose the first solution for mean estimation. Following [20], T. Nguyên propose *Harmony* for mean estimation and machine learning. Wang et al. [6] further propose hybrid mechanism *HM* for mean estimation and machine learning to improve *Harmony*. Ye et al. [25] focus on estimating the frequency and mean for key–value data. Sun et al. [26] propose *BiSample* for handling the missing data under LDP. Li et al. [27] propose to collect key–value data through distribution estimation.

The well-established Laplace mechanism [28] can also be adopted to guarantee to LDP [6]. Due to its high efficiency and effectiveness, we adopt it for adding noise. Moreover, we compare our *TESLA* with the state-of-the-art mean estimation and machine learning approaches mentioned above to demonstrate the generality of it.

2.3. Truth discovery based on cryptography

Truth discovery based on cryptography has also been widely studied. For example, the proposed approaches in [15,29,30] focus on not only the protection of workers' sensory data but also their reliability scores derived by the truth discovery approaches. [31] aim to alleviate the tremendous overhead incurred on the worker side. [32–34] completely remove the online requirement and tolerate workers offline at any stage. Zheng et al. [35] argue that the inferred truths of the requester should also be kept private and propose a protocol to tackle this problem. Following [35], Tang et al. [36] propose a more comprehensive protocol based on the data perturbation technology that can simultaneously protect the privacy of participants and truth results. Gao et al. [37] achieve data aggregation with high accuracy while preventing the leakage of both sensory data and tagged locations effectively. Liu et al. [38] design and implement a real-time privacy-preserving framework for sensory data streams. Xu et al. [39] design an approach enabling any entity to verify the correctness of aggregated results returned from the server.

These excellent works are all orthogonal to our work as we adopt LDP and do not involve any cryptography technologies.

2.4. Truth discovery based on local differential privacy

There are two broad settings for truth discovery based on local differential privacy, which are designed specifically for discrete answers and continuous answers according to the type of inputs.

Table 1
Summary and comparison.

Approaches	Discrete inputs	Continuous inputs	Rigorous LDP	Built-in noise
[8,10]	✓		✓	
[12]	✓			
[9,11]		✓		✓
[7]		✓		
TESLA		✓	✓	✓

For discrete answers, Li et al. [8] propose a two-layer approach, which can get a better utility in most cases. Sun et al. [10] propose a personalized incentive approach for binary-choice questions. Wang et al. [12] attempt to obtain the truths based on local attribute differential privacy under the assumption that only a portion of attributes are sensitive. It is a weaker version of LDP according to their statement.

For continuous answers, similar to [8], by adopting the Gaussian mechanism, Li et al. [9] propose a similar approach, which satisfies $(\epsilon,\ \delta)$-LDP with a certain probability. It is a weaker version of LDP. Sun et al. [7] propose a perturbation approach based on matrix factorization and try to perturb the latent factors of per worker. However, with the assumption that any pair of answer vectors differ by at most one cell, their approach satisfies ϵ-cell LDP, which is a weaker version of LDP. To tackle incentive problem, Sun et al. [11] propose a personalized incentive approach. Due to the adoption of the Gaussian mechanism, their approach satisfies ϵ-KL-LDP, which is still a weaker version of LDP.

We summarize them and compare them with our *TESLA* solution in Table 1.

To sum up, existing studies are either designed specifically for discrete inputs, or only satisfy the weaker versions of LDP for continuous inputs. Although directly adopting the Laplace mechanism for continuous inputs guarantees rigorous LDP, it may result in unreliable results as the contained noise could be excessive large. The reasons can be explained as follows. Due to the randomness and boundlessness of the Laplace distribution, the injected noise for privacy protection is likely to be extremely large even with a large ϵ. Moreover, there exists the built-in Gaussian noise within the worker-provided values, which may also have negative effect on the aggregated truths under LDP. Thus, all the above approaches may not get reliable utility for the real-world truth discovery under rigorous LDP with continuous inputs. To our best knowledge, we are the first to tackle this problem while guaranteeing rigorous LDP.

3. Preliminaries

In this section, we first describe the details about truth discovery. Then, we provide the details of local differential privacy, which will be used to obfuscate worker-provided values. Finally, the problem investigated in this paper is presented.

3.1. Truth discovery

Truth discovery is an effective way to obtain reliable results from conflicting data. *CRH* [13] is one of the most representative approaches currently, and does not consider privacy protection throughout the procedure. In particular, *CRH* consists of the following two phases:

- **Truth Aggregation**: In this step, each worker's weight is assumed to be known. Truth aggregation will be achieved by weighted summation of the provided values:

$$x_j^* = \frac{\sum_{i=1}^{M} w_i \cdot x_i^j}{\sum_{i=1}^{M} w_i}, \tag{1}$$

where M is the total number of workers; i is the index of the ith worker; x_i^j is the real value provided by the ith worker for the jth task; w_i is the weight for the ith worker, and x_j^* is the aggregated truth.

• **Weight Estimation**: In this step, the aggregated truths are fixed. Weight estimation is conducted based on the difference between the truths and the provided values:

$$w_i = -\ln \frac{\sum_{j=1}^{N} d\left(x_i^j, x_j^*\right)}{\sum_{i=1}^{M} \sum_{j=1}^{N} d\left(x_i^j, x_j^*\right)}, \quad (2)$$

where N is the number of tasks and $d(\cdot)$ is a function that measures the difference between the provided values and the aggregated truths. Different truth discovery methods may adopt various functions $d(\cdot)$, but the underlying principle is the same.

3.2. Local differential privacy

Local differential privacy (LDP) [5] is formally defined as follows.

Definition 1 (*ϵ-Local Differential Privacy*)**.** Given a privacy budget ϵ, a randomized algorithm A achieves ϵ-local differential privacy, iff for any two tuples t and $t' \in Dom(A)$, and for any possible $\tilde{t} \in Range(A)$, we have

$$\Pr\left(A(t) = \tilde{t}\right) \le e^{\varepsilon} \times \Pr\left(A\left(t'\right) = \tilde{t}\right),$$

where the probability is over the coin flips of A.

Typically, a classic mechanism for enforcing LDP is the Laplace mechanism [6].

Theorem 1 (*Laplace Mechanism*)**.** *For any function $f : D \to \mathbb{R}^n$ with sensitivity Δf, the algorithm*

$$A(D) = f(D) + < Lap_1(\lambda), \ldots, Lap_n(\lambda) >$$

satisfies ϵ-LDP, where $Lap_i(\lambda)$ is drawn i.i.d from a Laplace distribution with scale $\Delta f/\epsilon$. Δf is used to measure the maximum change in the outputs of a function when any individual's value is changed.

Laplace mechanism is also composable [28], including the sequential composition property and the post-processing property. In particular, they are:

Theorem 2 (*Sequential Composition*)**.** *Let $A_1, \ldots, A_k$ be k operations, each achieves ϵ_i-LDP. A sequence of operations $A_i(D)$ over database D achieves $\left(\sum \epsilon_i\right)$ -LDP.*

Theorem 3 (*Post-Processing*)**.** *Let A be an operation that is ϵ-LDP. Let B be an arbitrary operation acting on A. Then, composite operation $A \circ B$ is also ϵ-LDP*

3.3. Problem statement

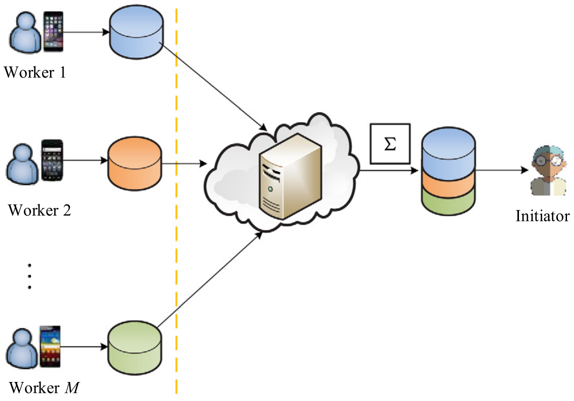


Fig. 1. Problem investigated.

Fig. 1 shows the investigated problem, which consists of four types of entities: M workers, N tasks, a server and an initiator. The server and M workers are interconnected through an internet environment. Each worker has performed a portion of the tasks. The server acts as a computing platform and conducts truth discovery using the uploaded noisy values. Finally, the aggregated truths are provided to the initiator.

Suppose the ith worker possesses a continuous value for the jth task x_i^j, where the value range of this task is $\left[d_{j1}, d_{j2}\right]$. The goal of this study is to design a solution to enable the server to obtain the noisy truth $\hat{x}_j^*$ from each worker's noisy value $\tilde{x}_i^j$ under a rigorous LDP model. For ease of presentation, we assume that all workers use the same privacy budget ϵ. During truth discovery, for any worker, an adversary could be the untrusted server, another worker or an outside attacker. Their objective is to learn a worker's true values.

Table 2 summarizes the notations that will be frequently used in this paper.

4. Our TESLA solution

4.1. Overview of TESLA

A first-cut solution for obtaining the aggregated truths under rigorous LDP is to conduct *CRH* based on the uploaded noisy values directly after adding Laplace noise. However, this strawman solution suffers from poor data utility. The root cause is that it is likely to incur markedly contained noise even with a large ϵ. The reasons can be explained as follows. According to the characteristics of the Laplace distribution, the injected noise for privacy protection is randomly sampled and unbounded, and this distribution drops markedly with respect to the x-axis. Moreover, there exists built-in Gaussian noise within the worker-provided values, which may also have a negative effect on the aggregated truths under LDP and makes the problem investigated in this paper far more challenging. Built-in Gaussian noise should also be carefully addressed to improve the accuracy of the inferred noisy truths.

To address these problems, we propose *TESLA* (**T**ruth discov**E**ry via probabilistic e**S**timation mall under rigorous **L**ocal differential priv**A**cy), whose key idea is to let the injected noise for privacy protection and the built-in Gaussian noise only weakly negatively affect the weight estimation and true aggregation. In particular, we estimate the supremum and infimum for each worker-provided noisy value to bound the noise. In particular, we estimate the supremum and infimum for each worker-provided noisy value to bound the contained noise. Based on the estimated supremum and infimum, we can obtain a more accurate fused noisy value to take the place of the worker-provided value. Subsequently, both weight estimation and true aggregation can benefit from this fused value. Moreover, we estimate workers' weight distributions by jointly considering the injected noise for privacy protection and the inherent Gaussian noise so as to be in favor of weight estimation. In particular, we first design a runtime filtering mechanism (*RFM*) to get the supremum and infimum of the noisy values. Then, a probabilistic fusion mechanism (*PFM*) to get the fused values. Finally, a probabilistic weight mechanism (*PWM*) is proposed, which serves as weight estimation of truth discovery.

Fig. 2 shows the overflow of *TESLA*. Specifically, *TESLA* mainly consists of the following phases:

Table 2
List of frequently used notations.

Notation	Definition
ϵ, ε	The total privacy budget and the privacy budget for a task
M, N	The total number of workers and tasks
N_i	The number of tasks the ith worker has done
x_{ij}, x_i	The true value of the ith worker for the jth task and the true value of the ith worker for a task
d_{j_1}, d_{j_2}	The minimum value and maximum value for the jth task
d_1, d_2	The minimum value and minimum value for a task
$\tilde{x}_{ij}$, $\tilde{x}_i$	The noisy value of the ith worker for the jth task and the noisy value of the ith worker for a task
d_{upper_ij}, d_{lower_ij}	One of upper bound and lower bound for $\tilde{x}_i^j$
d_{upper_i}, d_{lower_i}	One of upper bound and lower bound for $\tilde{x}_i$ of a task
d_{sup_ij}, d_{inf_ij}	The least upper bound of a set of d_{upper_ij} and the greatest lower bound of a set of d_{lower_ij}
d_{sup_i}, d_{inf_i}	The least upper bound of a set of d_{upper_i} and the greatest lower bound of a set of d_{lower_i}
$\hat{x}_{ij}$, $\hat{x}_i$	The fused value for $\tilde{x}_{ij}$ and $\tilde{x}_i$
$\hat{x}_{j*}$, $\hat{x}^*$	The aggregated noisy truth for the jth task and the aggregated noisy truth for a task
$P(\cdot)$, D	The probability value and the two-dimensional integral interval
θ, ρ	Server-specific comparison granularity and comparison probability

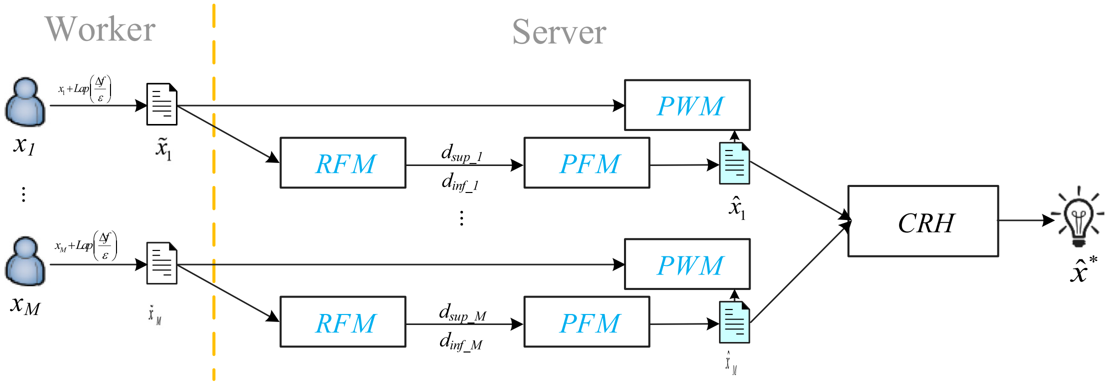


Fig. 2. Key steps of *TESLA* with their associated key techniques marked in blue.

Phase 1. Each worker obfuscates his value x_i by adopting the Laplace mechanism using privacy budget $\frac{\epsilon}{N_i}$ and sensitivity $\Delta f = d_2 - d_1$, where N_i is the number of tasks the ith worker has done. d_2 and d_1 are the maximum and minimum values for a certain task respectively. Then, he uploads the noisy $\tilde{x}_i$ to the server.

Phase 2. The server invokes *RFM* to get d_{sup_i} and d_{inf_i}, and invokes *PFM* to get the fused value $\hat{x}_i$. The details are presented in Section 4.2 and Section 4.3, respectively.

Phase 3. Based on the fused value $\hat{x}_i$, the server conducts truth discovery adopting *CRH* while using *PWM* as weight estimation, which is elaborated in detail in Section 4.4. Then, the aggregated truths are sent to the initiator.

In the following, we first provide a comprehensive analysis of *TESLA*, including its utility, privacy and complexity, in Section 4.5. Then, we present the details of *TESLA* about its extension for mean estimation, standard deviation calculation and multiple machine learning tasks in Section 4.6. Finally, we discuss *TESLA* in Section 4.7 to distinguish it from existing studies.

4.2. Runtime filtering mechanism

In *TESLA*, two types of noise are contained in the uploaded noisy values, which are the injected noise for privacy protection and the inherent Gaussian noise.

Since the Laplace distribution introduces noise with the variance of $2\lambda^2$ ($\lambda = \frac{d_2-d_1}{\epsilon}$), with the decreasing of privacy budget ϵ, the noise variance increases in a quadratic way. Moreover, according to the distribution characteristics of Laplace, the injected noise is unbounded, which is likely to cause great noise variance, even if ϵ is large. Similarly, the error by built-in Gaussian noise also increases in a quadratic way if the standard deviation is adopted to represent the accuracy of workers in non-privacy scenario [4]. Thus, we should mitigate the negative effect of the above two types of noise on weight calculation and truth aggregation together. For truth aggregation, we propose *RFM* and *PFM*. For weight calculation, we propose *PWM*.

Before providing the details of *RFM*, we first introduce two straightforward approaches.

One possible approach, called Laplace filtering (*LapFilter*), is to model the injected noise as much as possible. Another possible approach, called Gaussian filtering (*GauFilter*), is to model the inherent Gaussian noise as much as possible. Although *LapFilter* and *GauFilter* can achieve the proposed privacy goal as they work on noisy values, such approaches may not always lead to a satisfactory aggregated truth utility. The problem is that there is the injected noise and inherent Gaussian noise simultaneously after adding Laplace noise. We observe that these two straw-man approaches represent two extremes of data filtering: *LapFilter* only considers the injected noise, while *GauFilter* only considers the inherent Gaussian noise. This observation suggests the must considering of these two types of noise together.

Given a noisy value $\tilde{x}$ provided by a worker, we design *RFM* to get the fused value $\hat{x}$. The main idea of *RFM* is to define a probability comparison function according to the two types of noise to obtain the supremum and infimum for $\tilde{x}$. In what follows, we first state how to define the probability comparison function.

Let s_i and g_i denote the injected noise and inherent Gaussian noise, respectively, where $s_i \sim lap\left(\frac{d_2-d_1}{\epsilon}\right)$ and $g_i \sim N\left(\mu, \sigma^2\right)$. Thus, we have:

$$x_i = \tilde{x}_i - s_i - g_i.$$

Probability Comparison Function. Let v represent the value we need to compare with to get the supremum and infimum. Thus, we have:

$$\begin{aligned} P\left(x_i \leq v\right) &= P\left(\tilde{x}_i - s_i - g_i \leq v\right) \\ &= P\left(s_i + g_i \geq \tilde{x}_i - v\right) \end{aligned},$$

where $\tilde{x}_i$ and v are known by the server, and $P(\cdot)$ is the probability value. The above equation can be seen as a probability problem about two-dimensional continuous variables (s_i, g_i) in the plane set D, which indicates integral interval, denoted by:

$$D = \left\{(s_i, g_i) | s_i + g_i \geq \tilde{x}_i - v\right\}.$$

Let $f\left(s_i, g_i\right)$ denote the joint probability density function of (s_i, g_i). Since the variables s_i and g_i are independent from each other, we have

$$\begin{aligned} P\left(x_i \leq v\right) &= \iint_D f\left(s_i, g_i\right) ds_i dg_i \\ &= \int_{-\infty}^{+\infty} \left(\int_{-g_i+\tilde{x}_i^j-v}^{+\infty} f\left(s_i\right) f\left(g_i\right) ds_i\right) dg_i \end{aligned}, \tag{3}$$

where $f\left(s_i\right) = \frac{\varepsilon}{2(d_2-d_1)} e^{-\frac{|s_i|\varepsilon}{d_2-d_1}}$ and $f\left(g_i\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g_i-\mu)^2}{2\sigma^2}}$.

If $P\left(x_i \leq v\right) \geq \rho$, we can claim that one of the upper bounds for $\tilde{x}_i$ is v, where ρ is a server-specific probability. Similarly, if $P\left(x_i > v\right) \geq \rho$, we can claim that one of the lower bounds for $\tilde{x}_i$ is v. Note that, since Eq. (3) is monotonically increasing with the lower bound of the integral, the value range of the integral with regard to ρ is $[-g_i - d_2 + d_1, -g_i + d_2 - d_1]$. Additionally, the variance σ can be calculated based on the uploaded noisy values. According to the post-processing property of the Laplace mechanism (Theorem 3), it does not divulge privacy.

Binary Search based Boundary Calculation. Next, we need to get the supremum and infimum. To achieve this purpose, we define ρ-supremum and ρ-infimum as follows, where ρ is the probability of confidence.

Definition 2 (*ρ-Supremum*).

$$d_{sup\_i} = \min_{d_{upper\_i}} \left\{\Pr\left(d_{upper\_i} \leq v\right) \geq \rho\right\} \tag{4}$$

where $v \in [\, d_1, d_2]$ is the value we need to compare with, and d_{upper_i} is the upper bound estimated from the noise value $\tilde{x}_i$ of the ith worker. d_{sup_i} is the least upper bound of a set of d_{upper_i}.

Definition 3 (*ρ-Infimum*).

$$d_{inf\_i} = \max_{d_{lower\_i}} \left\{\Pr\left(d_{lower\_i} \geq v\right) \geq \rho\right\} \tag{5}$$

where d_{lower_i} is the lower bound estimated from the noise value $\tilde{x}$ of the ith worker. d_{inf_i} is the greatest lower bound of a set of d_{lower_i}.

We can vary v to obtain the possible upper bounds and lower bounds. Then, we set the minimum upper bound and the maximum lower bound as the supremum and infimum, respectively. However, since we are dealing with continuous values, there are infinite values of comparison granularity v. No matter how small v is, it will bring large information loss. Moreover, traversing all the values in $[d_1, d_2]$ may produce high computational cost. We find that we can utilize a binary search to control the comparison granularity v by setting the search stop condition θ.

Algorithm 1: RFM

```
Input: $d_1$ and $d_2$: the value boundaries for a task
Input: $\tilde{x}_i$: noisy value
Input: $\theta$: server-specific comparison granularity
Input: $\rho$: server-specific comparison probability
Output: $d_{sup\_i}$ and $d_{inf\_i}$: supremum and infimum for $\tilde{x}_i$
1:  Initialize $L = d_1$, $R = d_2$ and $d_{inf\_i} = L$;
2:  While $|R - L| \geq \theta$ Do
3:      $v = \frac{L+R}{2}$;
4:      If $p(x_i > v) \geq \rho$ Then
5:          $d_{inf\_i} = v, L = v$;
6:      Else
7:          $R = v$;
8:  Initialize $L = d_1$, $R = d_2$ and $d_{sup\_i} = R$;
9:  While $|R - L| \geq \theta$ Do
10:     $v = \frac{L+R}{2}$;
11      If $p(x_i \leq v) \geq \rho$ Then
12:         $d_{sup\_i} = v, R = v$;
13:     Else
14:         $L = v$;
15: Return $d_{sup\_i}$ and $d_{inf\_i}$.
```

Algorithm 1 shows the main steps of *RFM*. Specifically, we first invoke a binary search to obtain the supremum (Lines 1–7). Then, similarly, we get the infimum (Lines 8–14). Finally, we return d_{sup_i} and d_{inf_i} (Line 15). It follows the same manner for other workers' noisy values.

Parameters ρ and θ can be set empirically and experimentally as setting them does not hinder privacy. Specifically, *RFM* takes a post-processing on the server side, and the workers only connect with the server when uploading the noisy values.

4.3. Probabilistic fusion mechanism

After determining d_{sup_i} and d_{inf_i}, how to develop a fusion mechanism under the truth discovery scenario still remains a challenge. An intuitive method is to use the mean of the above two to replace each noisy value $\tilde{x}_i$. Let $\hat{x}_i$ represent the fused value, and we have:

$$\hat{x}_i^1 = \frac{d_{sup\_i} + d_{inf\_i}}{2}. \tag{6}$$

However, in this way, it may bring great information loss.

According to the spike property of the Laplace distribution, it is observed that the fusion utility can be improved by relying more on supremum d_{sup_i} or infimum d_{inf_i}. Thus, we propose two types of fusing methods:

$$\hat{x}_i^2 = d_{inf\_i} + \frac{\tilde{x}_i - d_{inf\_i}}{d_{inf\_i} - d_{inf\_i}} \cdot \left(\tilde{x}_i - d_{inf\_i}\right). \tag{7}$$

$$\hat{x}_i^3 = d_{inf\_i} - \frac{\tilde{x}_i - d_{inf\_i}}{d_{inf\_i} - d_{inf\_i}} \cdot \left(\tilde{x}_i - d_{inf\_i}\right). \tag{8}$$

To better understand them, we give Example 1.

Example 1. Suppose $x_i = 3$, $\tilde{x}_i = 2$, $d_{sup\_i} = 10$ and $d_{inf\_i} = 1$. Thus, we can get $\hat{x}_i^1 = 5.5$, $\hat{x}_i^2 = 1.11$ and $\hat{x}_i^3 = 9.89$. Clearly, $\hat{x}_i^2$ is the best. In addition, when $d_{sup\_i} = 3$ and $d_{inf\_i} = -6$, we can get $\hat{x}_i^1 = -1.5$, $\hat{x}_i^2 = -5.11$ and $\hat{x}_i^3 = 2.11$.

Therefore, *PFM* works as follows. If $\hat{x}_i - d_{inf\_i} \le d_{sup\_i} - \hat{x}_i$, that is, $p\left(x_i \le \frac{d_{sup\_i}+d_{inf\_i}}{2}\right) \ge \rho$, we use $\hat{x}_i^2$; otherwise, we use $\hat{x}_i^3$. In particular, $p\left(x_i \le \frac{d_{sup\_i}+d_{inf\_i}}{2}\right)$ is calculated by replacing v in Eq. (3) with $\frac{d_{sup\_i}+d_{inf\_i}}{2}$. In addition, if $d_{inf\_i} = d_1$ or $d_{sup\_i} = d_2$, it may introduce excessive information loss, canceling out the benefit of mitigating noise. In such a case, we do not conduct *PFM*.

Note that the methods in Eq. (7) and Eq. (8) can be viewed as special cases of the method in Eq. (6) if $\hat{x}_i$ exactly equals the mean. Technically, we can adopt any complex fusion method (e.g., compressed sensing), but the idea is also the same as *PFM*. Algorithm 2 shows the main steps of *PFM*.

```
Algorithm 2: PFM
Input: $\tilde{x}_i$: the noisy value of the $i$-th worker
Input: $d_{sup\_i}$ and $d_{inf\_i}$: the supremum and infimum for $\tilde{x}_i$
Input: $\rho$: server-specific comparison probability
Output: $\hat{x}_i$: the fused value
If $d_{inf\_i} = d_1$ or $d_{sup\_i} = d_2$ Then
    Return $\tilde{x}_i$;
Else
    If $p\left(x_i \le \frac{d_{sup\_i}+d_{inf\_i}}{2}\right) \ge \rho$ Then
        Get $\hat{x}_i$ via Eq. 7;
    Else
        Get $\hat{x}_i$ via Eq. 8;
Return $\hat{x}_i$.
```

To quantify the fused utility of *PFM*, we adopt the following utility definition [9].

Definition 4 (α, β-*Accuracy*)**.** Let $\beta \in [0, 1]$ and $\alpha \ge 0$, a method B satisfies (α, β)-*accuracy* if the following inequality holds:

$$\Pr\left[\left|\tilde{x} - B\left(\tilde{x}\right)\right| \ge \alpha\right] \le \beta.$$

Intuitively, this definition indicates that the fused utility is larger than α with the probability at most β. Clearly, a smaller α means better fusion utility under a certain β. Thus, we derive the quantitative relationship before and after adopting *PFM* as the following theorem.

Theorem 4. *Utility of PFM. For a given β, the fused utility of PFM can be found to be:*

$$\alpha = \sqrt{\frac{2\left(\frac{d_2-d_1}{\varepsilon}\right)^2 + \sigma^2}{\beta}}.$$

Proof. The fused utility of *PFM* can be written as

$$PFM\left(\tilde{x}\right) - \tilde{x} = \hat{x} - \tilde{x} = \eta,$$

where $\eta = s + g$. In addition, s is the added Laplace noise, and g is the inherent Gaussian noise.

Recall that the variance of added Laplace noise is $D(s) = 2\left(\frac{d_2-d_1}{\varepsilon}\right)^2$, and the variance of inherent Gaussian noise is $D(g) = \sigma^2$, we can derive that,

$$D(\eta) = 2\left(\frac{d_2-d_1}{\varepsilon}\right)^2 + \sigma^2.$$

Therefore, from the Chebyshev's inequality, we have

$$\Pr\left[\left|\tilde{x} - PFM\left(\tilde{x}\right)\right| \ge \alpha\right] \le \frac{2\left(\frac{d_2-d_1}{\varepsilon}\right)^2 + \sigma^2}{\alpha^2}.$$

Thus, for a given β, we have $\alpha = \sqrt{\frac{2\left(\frac{d_2-d_1}{\varepsilon}\right)^2+\sigma^2}{\beta}}$, which indicates that the fused utility satisfies the $\left(\alpha, \frac{2\left(\frac{d_2-d_1}{\varepsilon}\right)^2+\sigma^2}{\alpha^2}\right)$-accuracy.

4.4. Probabilistic weight mechanism

Inspired by Gaussian mixture model for modeling mixed error, we can formally define the total error distribution as follows:

$$p(x) = \frac{1}{2} \cdot \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{9}$$

where $2\lambda^2$ is the variance of the Laplace distribution and σ is the standard deviation of the Gaussian distribution.

An intuitive method is to directly apply the existing weight estimation formula, as shown in Eq. (2), to estimate each worker's weight, where $d(\cdot)$ can be measured by *Euclidean Distance* or *Manhattan Distance*. However, due to the fact that this intuitive weight estimation scheme does not consider the above two types of noise, it will lead to inaccurate obtained weights. Since weight plays the leading role in truth discovery, we may not get reliable aggregated truths under rigorous LDP.

To address this issue, since the aggregated truth can be greatly improved by distinguishing high-quality workers from the others and relying on these identified high-quality workers, we design *PWM* by considering both types of noise to mitigate their negative influence on weight estimation. The details of *PWM* are shown as follows.

Let x represent the value without noise and $\tilde{x}$ represent the value after adding Laplace noise locally, we have:

$$x - \tilde{x} \sim p(x).$$

The common principle of truth discovery is that a worker will be assigned a higher weight if his provided values are closer to the aggregated truths, and the provided values of this worker will be counted more in the aggregation phase if he has a higher weight. Following this principle, we formulate a constrained nonlinear programming problem to obtain a new weight estimation scheme, which will serve as the weight estimation of *CRH*.

$$\sum_{i=1}^{M} \sum_{j=1}^{N} w_i \left(\frac{1}{2\lambda} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(\hat{x}_i^j - \hat{x}_j^*\right)^2}{2\sigma^2}} \right). \tag{10}$$
$$s.t.\ \ w_i^2 = 1$$

The above objective function measures the weighted distance between the provided value $\hat{x}_i^j$ and the aggregated truth $\hat{x}_j^*$. Specifically, if a worker-provided value is far from the aggregated truth, to minimize the total loss, it will be assigned a low weight. Thus, the aggregated truth will be closer to the values from high-quality workers. That is, *PWM* also exactly follows the general principle of truth discovery.

Since an arbitrary norm on $\mathbb{R}^n$ is convex, non-negative summation is convex-preserving operations, and the formulated problem can be solved by the Lagrange multipliers approach.

The Lagrangian of Eq. (10) is given as,

$$L\left(w_i, \lambda\right) = \sum_{i=1}^{M}\sum_{j=1}^{N} w_i \left[\frac{1}{2\lambda} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|^2}{2\sigma^2}} \right], \\ +\zeta \left(\sum_{i=1}^{M} w_i^2 - 1 \right)$$

where ζ is a Lagrange multiplier. Let the first-order derivative of Lagrangian with respect to w_i be 0, we can get:

$$2\zeta w_i = \sum_{j=1}^{N} \left(\frac{1}{2\lambda} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|^2}{2\sigma^2}} \right). \tag{11}$$

From the constraint that $w_1^2 + \cdots + w_i^2 + \cdots + w_M^2 = 1$, we can derive that:

$$\zeta = \frac{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} \left[\frac{1}{2\lambda} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left|\hat{x}_i^j - \hat{x}_j^*\right|^2}{2\sigma^2}} \right]}}{2}. \tag{12}$$

By plugging Eq. (12) into Eq. (11), we can get:

$$w_i = \frac{\frac{1}{2\lambda} e^{-\frac{\left|\hat{x}_i - \hat{x}^*\right|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left|\hat{x}_i - \hat{x}^*\right|^2}{2\sigma^2}}}{\sqrt{\sum_{i=1}^{M} \left[\frac{1}{2\lambda} e^{-\frac{\left|\hat{x}_i - \hat{x}^*\right|}{\lambda}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left|\hat{x}_i - \hat{x}^*\right|^2}{2\sigma^2}} \right]}}, \tag{13}$$

where $\lambda = \frac{d_{2j} - d_{1j}}{\epsilon}$ and σ can be calculated based on the uploaded noisy values.

4.5. Analysis of *TESLA*

Privacy Guarantee. The following theorem establishes the privacy guarantee of *TESLA*.

Theorem 5. *Our TESLA solution satisfies rigorous ϵ-LDP.*

Proof. In *TESLA*, the server needs to access each worker's raw data exactly once. That is, in **Phase 1**, each worker invokes the Laplace mechanism using privacy budget ϵ to generate his noisy data. ϵ is divided into $\frac{\epsilon}{N_i}$ parts for each worker, where N_i is the number of tasks the ith worker has done. By the sequential composition property in Theorem 2, the noise addition operation achieves ϵ-LDP. Note that, since the subsequent *RFM* and *PFM* do not require the server to access any worker's raw data, there will not lead to any privacy risk or consume any privacy budget. Additionally, to obtain the noisy truths, *CRH* is conducted on already locally differentially private outputs. In particular, they are fused data generated by *PFM*. By the post-processing property in Theorem 3, *TESLA* can guarantee rigorous ϵ-LDP for each worker. This completes the proof.

Utility Analysis. We have the following utility theorem for *TESLA*.

Theorem 6. *The expected error of an aggregated truth before and after adopting TESLA satisfies the following inequality:*

$$\mathbf{E}\left[\left|x^* - \hat{x}^*\right|\right] \le \sqrt{\frac{2}{\pi}}\sigma + \frac{d_2 - d_1}{\varepsilon}.$$

Proof. Let $\mathbf{E}$ and E denote the error and expectation respectively. We have,

$$\begin{aligned} &\mathbf{E}\left[\left|x^* - \hat{x}^*\right|\right] \\ &= \mathbf{E}\left[\left|\frac{\sum_{i=1}^{M} w_i x_i}{\sum_{i=1}^{n} w_i} - \frac{\sum_{i=1}^{M} W_i \hat{x}_i}{\sum_{i=1}^{n} W_i}\right|\right] \\ &= \mathbf{E}\left[\left|\frac{\sum_{i=1}^{M} w_i x_i \sum_{i'=1}^{M} W_{i'} - \sum_{i=1}^{M} w_i \sum_{i'=1}^{M} W_i \hat{x}_{i'}}{\sum_{i=1}^{M} w_i \sum_{i'=1}^{M} W_{i'}}\right|\right] \end{aligned}.$$

Since $\sum_{k=1}^{M} a_k b_k \sum_{k'=1}^{M} c_{k'} = \sum_{k=1}^{M} \sum_{k'=1}^{M} a_k b_k c_{k'}$, we have,

$$\begin{aligned} &\mathbf{E}\left[\left|x^* - \hat{x}_i^*\right|\right] \\ &= \mathbf{E}\left[\left|\frac{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'} x_i - \sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'} \hat{x}_{i'}}{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'}}\right|\right] \end{aligned}.$$

Moreover, since $E\left[|X+Y|\right] \le E\left[|X|\right] + E\left[|Y|\right]$, we have,

$$\begin{aligned} &\mathbf{E}\left[\left|x^* - \hat{x}_i^*\right|\right] \\ &\le \frac{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'} E\left[\left|x_i - \hat{x}_{i'}\right|\right]}{\sum_{i=1}^{n}\sum_{i'=1}^{n} w_i W_{i'}} \\ &= \frac{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'} E\left[|g_i + s_i|\right]}{\sum_{i=1}^{n}\sum_{i'=1}^{n} w_i W_{i'}} \\ &\le \frac{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'} \left(E\left[|g_i|\right] + E\left[|s_i|\right]\right)}{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'}} \end{aligned}.$$

Furthermore, since $E\left[|g_i|\right] = \sqrt{\frac{2}{\pi}}\sigma$ and $E\left[|s_i|\right] = \frac{d_2 - d_1}{\varepsilon}$, we have

$$\begin{aligned} &\mathbf{E}\left[\left|x^* - \hat{x}_i^*\right|\right] \\ &\le \frac{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'} \left(\sqrt{\frac{2}{\pi}}\sigma + \frac{d_2 - d_1}{\varepsilon}\right)}{\sum_{i=1}^{M}\sum_{i'=1}^{M} w_i W_{i'}} \\ &\le \frac{\sum_{i=1}^{M}\sum_{i'=1}^{M} \left(\sqrt{\frac{2}{\pi}}\sigma + \frac{d_2 - d_1}{\varepsilon}\right)}{M^2} \\ &= \sqrt{\frac{2}{\pi}}\sigma + \frac{d_2 - d_1}{\varepsilon} \end{aligned}.$$

Complexity. For communication complexity, the server merely communicates with each worker when he uploads the noisy value $\tilde{x}$. Since there are M works and N tasks, the total communication complexity is $O(MN)$.

For time complexity, there are four parts of computation. The first is adding noise locally. Since there are M works and N tasks, it leads to totally $O(MN)$. The second is invoking *RFM*. Recall that the server needs to get d_{sup_i} and d_{inf_i} for each noisy value through twice binary search, and there are $\frac{d_2 - d_1}{\theta}$ values to be compared with the server-specific comparison granularity. Since the complexity of binary search is $O(\log Q)$, where Q is the amount of values to be sorted, *RFM* takes a total of $2O\left[\log\left(\frac{d_2 - d_1}{\theta}\right)\right]$. The third is invoking *PFM*. Since there are MN values to be processed, and one comparison operation taking $O\left(1\right)$ is required for each processing, *PFM* uses a total of $O(MN)$. The fourth is conducting truth discovery using Eq. (13) and Eq. (1). According to [15], it consumes a total of $n \cdot O\left(2MN\right)$, where n is the number of iterations. Thus the total time complexity is $2O(MN)$ + $2MN \cdot O\left[\log\left(\frac{d_2 - d_1}{\theta}\right)\right]$+ $n \cdot O(2MN)$.

In general, it indicates that the communication complexity and computation complexity of *TESLA* is linear w.r.t. the number of observations. It is easy to implement and use in real practice, which makes it a good choice in practical applications.

4.6. Extending *TESLA* for other data analysis tasks

Since *TESLA* relies only on the properties of Laplace noise to satisfy LDP, it is independent of the potential truth discovery approaches. We use *TESLA* only to obtain a better version of the noisy values, and extend it to various data analysis tasks where continuous values should be protected.

We extend *TESLA* for mean estimation and standard deviation calculation, which are two typical tasks for continuous values under rigorous LDP. Specifically, we compute the mean with our fused values, and compare it with several state-of-the-art methods. We compute the standard deviation by using the computed mean and our fused values.

We extend *TESLA* to various machine learning tasks, such as support vector machine (*SVM*), neural network, logistic regression and bayesian network. In particular, we use the filtered values to conduct tasks or calculate the required distribution.

Note that, we also verify whether the idea of *TESLA* can be used to improve the existing studies.

4.7. Discussion of TESLA

Existing studies are either designed specifically for discrete inputs, or only satisfy the weaker versions of LDP for continuous inputs. Although directly adopting for continuous inputs guarantees rigorous LDP, it may result in unreliable results as the contained noise could be excessively large. To demonstrate the superiority of *TESLA*, we present the properties of it that distinguish it from existing approaches. It takes a post-processing approach on the server side by elaborating the inherent noise and the injected noise. In particular, first, the workers perturb their true data with the Laplace mechanism and send it to the server. Then, the server runs the following three procedures: (1) estimating the supremum and infimum of the perturbed value from each worker using proposed *RFM*; (2) estimating each worker's true values by fusing the corresponding estimated supremum and infimum using the proposed *PFM*; and (3) estimating workers' weight distributions using the proposed *PWM*.

Our *TESLA* solution is different from existing approaches mainly in the following aspects:

- **Non-trivial solution under Rigorous LDP**. Since the weak versions of LDP have a higher probability of be attacked, it is important to have rigorous LDP. Intuitively, we may process the discrete data into continuous data and adopt existing LDP approaches for discrete data to achieve rigorous LDP. However, the utility could be very poor even if we adopt different encoding methods, which has also been proven by [7]. The reason is that the value of each worker may fluctuate marginally near the aggregated truth, and no matter what encoding methods are adopted, it will bring great information loss. Moreover, too small coding granularities will lead to large domain problems in LDP [6]. Furthermore, intuitively, we may add Laplace noise instead of using the Gaussian mechanism to achieve rigorous LDP. Due to the randomness and boundlessness of the Laplace distribution, it will result in unreliable results. Thus, we can only resort to more accurate noisy values. Therefore, these factors jointly derive the interrelated solution in this paper.
- **Built-in noise**. For the first time, we explore the built-in noise that widely exists in various crowdsensed data under rigorous LDP, which was systematically overlooked in the recent studies.
- **Stronger Privacy with Excellent performance by the only cost of time**. We know that privacy and utility have certain trade-off, that is. Thus, we need to sacrifice some utility to achieve stronger privacy. In this submission, we achieve stronger privacy while showing improved utility. As shown in the procedures of *TESLA*, the only cost is time. Specifically, the server spends most of the time on estimating the supremum and infimum by *RFM*. Although such estimations require more computation on the server side, they are still efficient as the computational complexity is linearly related to the amount of data according to the discussion about complexity. Moreover, in practice, it is reasonable to assume that the server has strong computing power to perform the estimations. Furthermore, we notice that the number of values is normally not unreasonably large due to crowdsensing economic budget. Besides, having a more powerful machine and making use of distributed computing can further improve the runtime. As such, we believe that our solution can provide an acceptable trade-off for real-world deployments.
- **Easy Extension for Other Truth Discovery Scenarios.** In *TESLA*, we can usually get a good estimated value on the premise of knowing the noise distribution. Thus, we can easily extend it to other scenarios [40,41]. For <*location*, *data*> scenario [42], since there usually exist certain correlations between the location and data, it may lead to the disclosure of private data as the private data could be inferred by the correlated non-private data [43,44]. For example, the air quality in one place is excellent, the air quality in other places is poor. If only the location is protected, the attacker can infer the location information according to the air quality level, which result in privacy disclosure [45]. Therefore, both location and data need to be protected. Intuitively, we may directly add noise to both location and data, which may destroy the correlation between them and negatively affect the utility of truth discovery. We have noticed that in location-based service, a series of excellent achievements can be used for Ref. [46–48]. For example, we may learn from [46], and generate a series of interrelated false requests for each worker's data. Then these false requests are fused in a certain way (such as weighted average). Next, a method similar to *RFM* is used for filtering, and a method similar to *PFM* is used to redesign for weight updates. Finally, we submit the noisy <*location*, *data*> to the server for truth value discovery. Additionally, in some scenarios, what tasks have been done may be also sensitive. We may carry out scheme design in combination with [41,42].
- **Easy Extension for Other Data Analysis Tasks**. Due to the post-processing property of *TESLA*, it in the field of LDP is somewhat like the role of homomorphic encryption in *SMC* (Secure Multi-party Computation) [30]. Actually, almost all the privacy protection scenario dealing with continuous data can adopt our idea or even solution directly. Indeed, we have performed various comparative experiments for its applicability, which are shown in Section 5.4.

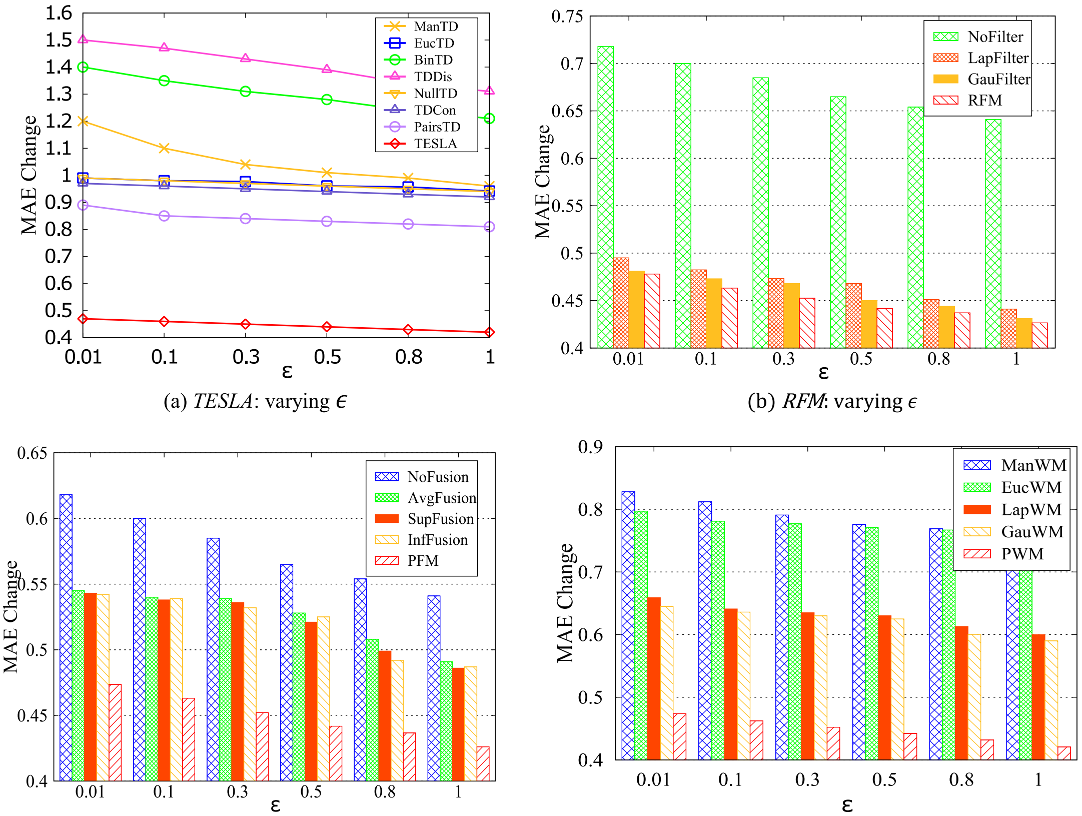
5. Experiments

In this section, we first evaluate *TESLA* and its core building blocks, including *RFM*, *PFM* and *PWM*, by varying the privacy budget ϵ over two real-world datasets to verify their effectiveness. Then, we further evaluate the core building blocks by varying different parameters, such as the number of worker-provided values M, the server-specific comparison granularity θ and the server-specific comparison probability ρ. Next, we verify *TESLA* by mean estimation and standard deviation calculation over a synthetic dataset, which are two typical tasks for continuous inputs under LDP. Finally, we extend *TESLA* for various machine learning tasks over another large-scale generated synthetic dataset.

5.1. Experimental setup

Datasets. We use two real-world datasets and two synthetic datasets to comprehensively evaluate *TESLA*.

- **Adult Content**: It [7], *Adu* for short, is collected from thousands of websites about their received scores, which range from 1 to 5. There are 89,796 claims from 825 workers on 11,040 tasks.
- **Weather**: It [13], *Wea* for short, contains the statistics of weather data from 30 cities in the United States from Jan. 28, 2010 to Feb. 4, 2010. Following the common sense [49], we consider the information from Accuweather.com to be the ground truths. There are 16,038 claims from 1920 workers on 1740 tasks.



(a) *TESLA*: varying ϵ

(b) *RFM*: varying ϵ

(c) *PFM*: varying ϵ

(d) *PWM*: varying ϵ

Fig. 3. Performance Comparison over *Adu*.

- **Synthetic dataset 1**: We generate 1000 values in [1,10], to create *Syn1* and set its ground truth to be 5.5 for evaluation. Similar to [7], for each worker, his answer is generated by adding the Gaussian noise N (5.5, σ_i^2) to the ground truth, where σ_i is decided by the worker's quality. We define three types of workers: high-quality workers with $\sigma_i = 1$, middle-quality workers with $\sigma_i = 5$ and low-quality workers with $\sigma_i = 10$. We randomly choose 20% of the workers to be high-quality workers, 60% of the workers randomly to be middle-quality workers, and the rest to be low-quality workers.
- **Synthetic dataset 2**: It, *Syn2* for short, contains 250,000 records, each of which contains 30 attributes, and the threshold range of each attribute is 20, where the last attribute has 8 values and is represented as 8 categories.

Evaluation Metrics and Baselines. For the utility measure of truth discovery, we adopt the commonly used *MAE Change* (Mean Absolute Error Change) [7], which measures the deviation between the noisy truths and the real truths, where a smaller value indicates better performance. Moreover, we also consider the running time of *TESLA* as another evaluation metric. For the utility measure of classification tasks in machine learning, we adopt the commonly used accuracy *Acc*, where a higher value indicates better performance. For the utility measure of bayesian network task in machine learning, we adopt the commonly used total variation distance *TVD* [50], which measures the noisy marginal and noise-free marginal. The smaller, the better.

For the baselines, we compare *TESLA* with several state-of-the-art approaches. Specifically, we compare it with *TDDis* (Truth Discovery for DIScrete inputs) [8], *TDCon* (Truth Discovery for CONtious inputs) [9], *PairTD* (Pair inputs for Truth Discovery) [11], *BinTD* (Binary inputs for Truth Discovery) [10], *NullTD* (Null privacy for Truth Discovery) [7]. Note that, since *TDDis* and *BinTD* are designed for discrete inputs, we discretize the continuous data to fit them. Moreover, we also compare *TESLA* with the designed approaches *HamTD* and *EucTD*, where *CRH* is conducted based on the uploaded noisy values when $d(\cdot)$ in Eq. (2) is measured by Hamming Distance or Euclidean Distance respectively.

For machine learning and mean estimation, we compare the variant of *TESLA* with the state-of-the-art *Harmony* [51], *HM* (Hybrid Mechanism) [6] and *CKV* (Collecting Key–Value data) [27].

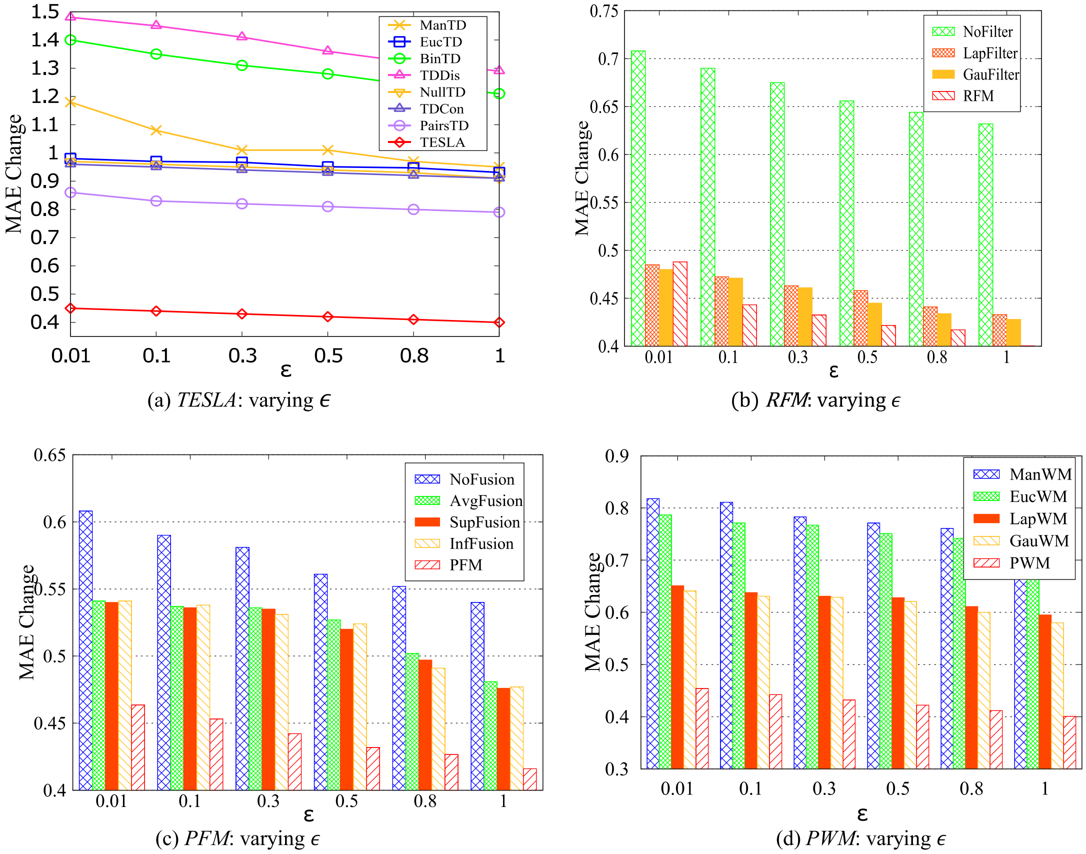
We implement all approaches in Python 3.7. All experiments are conducted on an Intel i5-5200U 2.20 GHz laptop.

5.2. Performance comparison

Impact of ϵ. Figs. 3(a) and 4(a) show the performance of *TESLA* when varying ϵ. We have the following observations. *TESLA* outperforms all competitors in all cases. This is because it mitigates the negative influence of the noise contained in the uploaded noisy values. Note that, *TDCon*, *NullTD* and *PairsTD* only guarantee the weaker versions of LDP. Even with these privacy relaxations, *TESLA* still significantly outperforms them.

Effectiveness of RFM. We compare *RFM* with *LapFilter* and *GauFilter*, which only consider the injected the Laplace noise or the inherent Gaussian noise respectively. Moreover, we also include *NoFilter* to show the utility improvement after filtering. Figs. 3(b) and 4(b) show the results. We have the following observations. First, the methods considering filtering the noisy values, significantly outperform *NoFilter*. There are two types of noise in the noisy values. By limiting the scale of the noise, the utility can be improved. Second, *RFM* always performs the best when ϵ gradually enlarges. This is because by utilizing the defined probability comparison function, we can limit the scale of the noise to some extent.

Effectiveness of PFM. To verify the effectiveness of *PFM*, we compare it with four baselines, which are *NoFusion*, *AvgFusion*,



(a) *TESLA*: varying ϵ

(b) *RFM*: varying ϵ

(c) *PFM*: varying ϵ

(d) *PWM*: varying ϵ

Fig. 4. Performance Comparison over *Wea*.

SupFusion and *InfFusion*. For *NoFusion*, the server directly conducts *CRH* on the noisy values. For *AvgFusion*, the server adopts the scheme in Eq. (6) to filter the noisy values. For *InfFusion* and *SupFusion*, the server adopts the schemes in Eq. (7) and Eq. (8) respectively.

Figs. 3(c) and 4(c) show the results. We have the following observations. First, the methods considering fusing the obtained supremum (d_{sup_i}) and infimum (d_{inf_i}), significantly outperform *NoFusion*. Second, *InfFusion* and *SupFusion* perform better than *AvgFusion*. This is because, in most cases, the true value is closer to supremum or infimum, and *AvgFusion* cannot preserve sufficient information, which inevitably leads to poor performance. Third, *PFM* always performs the best when ϵ gradually increases. The reason is by adaptively relying more on the supremum or infimum, information loss can be reduced as much as possible.

Effectiveness of PWM. To verify the effectiveness of *PWM*, we compare it with four baselines, which are *HamWM*, *EucWM*, *LapWM* and *GuaWM*. For *HamWM* and *EucWM*, they use Hamming Distance and Euclidean Distance for $d(\cdot)$ in Eq. (2) respectively. For *LapWM* and *GuaWM*, they set the noise distribution in Eq. (9) as the Laplace distribution and the Gaussian distribution respectively.

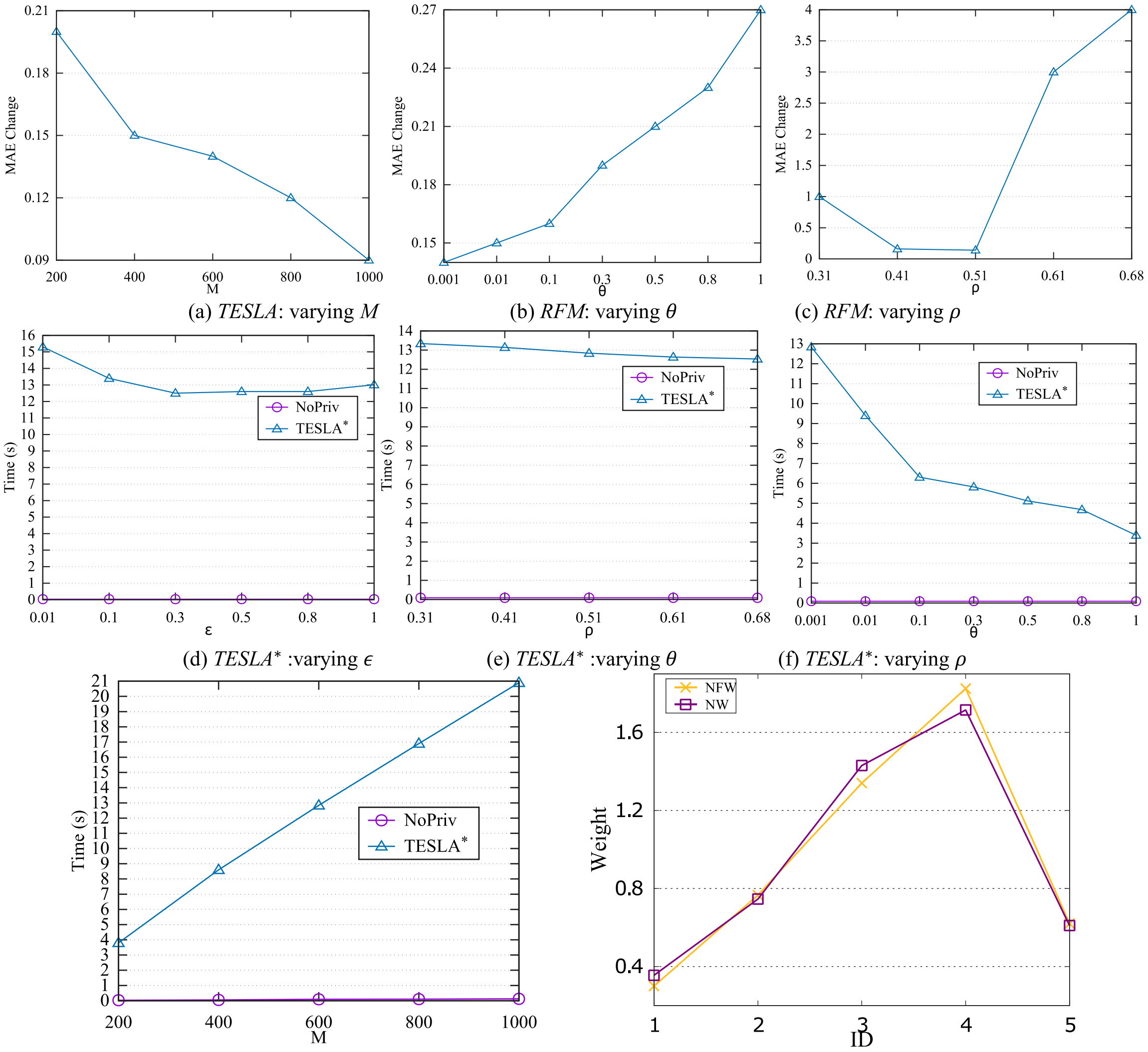
Figs. 3(d) and 4(d) show the results. Our findings are two-fold. First, the methods considering the noise significantly outperform *HamWM* and *EucWM*. The reason lies in that, by modeling the noise for the uploaded noisy values, we can reduce the negative impact of noisy values on truth aggregation and weight estimation. Second, *PWM* always performs best when ϵ gradually increases. This is because the above two types of noise exist simultaneously in the uploaded values, and considering them together can make the potential truth discovery approaches (e.g., *CRH*) more robust against noise to the maximum extent.

5.3. Effect of different parameters

Impact of *M*. Fig. 5(a)shows the results when varying the number of workers M. It witnesses a sharp drop of the *MAE Change* while decreasing M. The reason is that *CRH* can estimate workers' weights better when more information is available.

Impact of θ and ρ. Figs. 5(b) and 5(c) show the results when varying the server-specific comparison granularity θ and server-specific comparison probability ρ. We have the following observations. First, a smaller θ means better performance. The reason is that with a small θ, the server can get tighter supremum and infimum, which can mitigate information loss due to privacy protection as much as possible. Second, the range of ρ in Eq. (3) is [0.3159, 0.6840] according to our analysis in subsection for *RFM*. We thus empirically set $\rho = 0.51$ as it always produces a relatively good performance. The reason is as follows. When ρ is too small, almost all the values will be fused, which leads to large information loss. When ρ is too large, the obtained supremum and infimum are close to the numerical boundaries d_1 and d_2, which will lead to greater fusion error. Third, the results change significantly, which indicates that *RFM* is easy to be trained. It is very important for practical deployment.

Efficiency of TESLA. Figs. 5(d), 5(e), 5(f) and 5(g) depict the execution time, where *NoPriv* represents the running time of *CRH* on the original values. Note that, since the adopted Scipy library may be non-convergent when computing a double integral, it requires nearly 3800s for *TESLA*. This comparison is meaningless due to the limitation of Python itself. We can use other programming languages to solve this problem. Hence, we only consider the injected Laplace noise in Eq. (3) (denoted by *TESLA**) to test the time cost varying ϵ, θ, ρ and M. We can observe that the running time after perturbation is a little larger than that on original



(a) *TESLA*: varying M (b) *RFM*: varying θ (c) *RFM*: varying ρ

(d) *TESLA** :varying ϵ (e) *TESLA** :varying θ (f) *TESLA**: varying ρ

(g) *TESLA** :varying M (h) *TESLA*:varying workers' id

Fig. 5. Effect of different parameters over *Syn*1.

values, and the running time changes little when ϵ varies. However, obviously, the running time is still acceptable. Moreover, the running time decreases when θ and ρ increases. Besides, the running time varying M is approximately linear, which is consistent with our complexity analysis. This guarantees practical deployment of *TESLA* on large-scale crowdsourcing applications. Furthermore, the complexity analysis of *TESLA* also confirms its scalability.

Impact on Weight Estimation. We randomly choose 5 workers to show the impact of *TESLA* on Weight Estimation before and after adopting *TESLA*. Fig. 5(h) shows the results, where *NFW* is the noise-free weight and *NW* is the noisy weight. We find that the weights from these two methods are approximately the same. The reason is noise-free values are obtained as far as possible to some extent. Since weight plays the leading role in truth discovery, we can expect the desirable performance using *TESLA*.

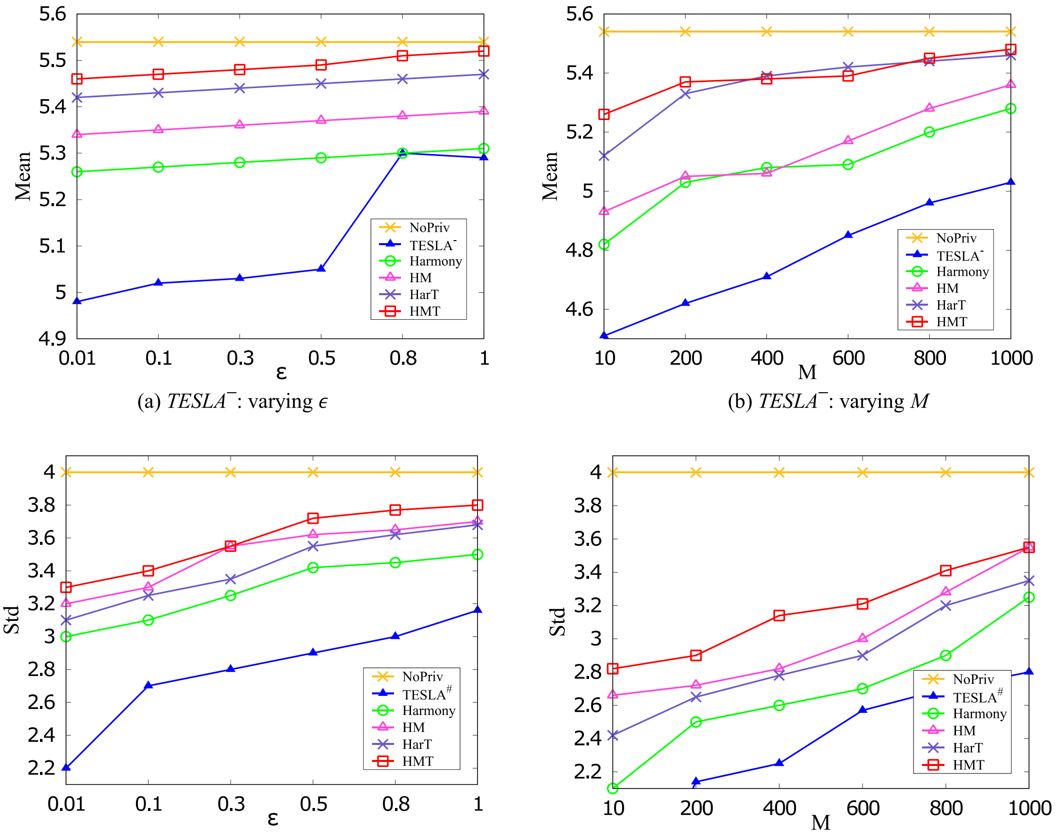
5.4. Performance on other data analysis tasks

Impact on Mean. We compare the generalized method *TESLA*$^{-}$, which does not conduct *PWM* in *TESLA*, for mean estimation with the state-of-the-art methods *Harmony* [51] and *HM* [6]. *HarT* is a variant of [6] Harmony when preprocessing data using *RFM* and *PFM*, and the same for *HMT* of *HM*. Fig. 6(a) and 6(b) shows the results when varying ϵ and M. It can be observed that *HarT* and *HMT* are always closer to *NoPriv* especially when ϵ and M are small. It demonstrates the effectiveness of the idea of *TESLA* as the quality of inputs is fundamentally improved. Moreover, there are obvious upward trends and the performance gap shrinks while enlarging ϵ and M. Thus, the idea of *TESLA* is more announced with small ϵ and M.

Impact on Standard Deviation. To eliminate the effect of randomness, we also test the calculation of standard deviation (Std) when varying ϵ and M, and the results are shown in Figs. 6(c) and 6(d). We calculate Std while the mean and each value are calculated by *TESLA*$^{-}$, and our method is denoted by *TESLA*$^{\#}$. We have the similar observations with mean estimation, and the reasons are the same.

Table 3 shows the results about different machine learning tasks on *Syn2* dataset. Due to space limitation, we only compare *TESLA*$^{+}$ with *HM* and *CKV*, which are the state-of-the-art method for machine learning under LDP. In particular, Bayes is bayes network, which is measured by *TVD*; *NN* and *LR* are neural network and logistic regression respectively, which, as well as *SVM*, are measured by Acc. We set different privacy budgets and data volumes for reporting the results, and have the following observations.

First, it can be seen that *HM* or *CKV* can be improved by *TESLA*$^{+}$, as *HMT* performs better than *HM* and *CKV*. Second, the



(a) *TESLA*$^{-}$: varying ϵ

(b) *TESLA*$^{-}$: varying M

(c) *TESLA*$^{\#}$: varying ϵ

(d) *TESLA*$^{\#}$: varying M

Fig. 6. Mean and variance estimations over *Syn*1.

Table 3
Acc (%) and TVD (%) of machine learning tasks over *Syn2*.

Tasks	Methods	$\epsilon = 8$		$\epsilon = 5$		$\epsilon = 4$		$\epsilon = 2$	
		100%	50%	100%	50%	100%	50%	100%	50%
Bayes	*TESLA*$^{+}$	79.115	77.532	75.324	71.814	74.568	72.856	72.156	70.526
	CKV	80.135	78.625	76.526	73.362	75.136	73.456	73.214	71.421
	HM	81.258	79.332	78.116	75.236	76.452	74.152	74.325	72.321
	HMT	83.366	80.784	80.532	77.145	77.856	75.235	75.541	73.452
NN	*TESLA*$^{+}$	85.051	82.133	83.232	78.942	81.263	76.456	79.325	77.562
	CKV	85.256	82.568	83.756	79.154	82.123	77.452	80.132	78.145
	HM	86.256	83.260	84.112	82.165	83.357	78.123	81.235	79.112
	HMT	88.166	85.623	85.652	83.845	84.421	79.236	82.356	80.156
LR	*TESLA*$^{+}$	83.388	81.790	79.214	76.720	78.135	75.153	76.135	75.132
	CKV	84.561	82.156	80.516	77.421	79.256	76.524	77.235	76.523
	HM	85.361	83.140	81.325	78.235	80.423	77.358	78.541	77.512
	HMT	87.147	85.231	83.561	80.265	81.236	78.254	79.154	78.541
SVM	*TESLA*$^{+}$	83.568	81.804	81.568	79.168	80.154	77.235	78.325	76.162
	CKV	83.896	82.165	81.886	79.352	81.352	78.132	79.421	77.341
	HM	84.220	82.725	82.425	80.326	82.145	79.521	80.132	78.642
	HMT	86.541	85.006	84.562	80.524	83.256	80.135	81.236	79.741

performance does not drop as quickly with decreasing data volume compared with the privacy budget. The reason can be explained as follows. More data contains more effective information, and the benefits of capturing more effective information outweigh the advantage of less noise. Third, there are obvious upward trends while enlarging ϵ and M.

6. Conclusion

In this paper, we focus on inferring truth effectively under rigorous local differential privacy for continuous inputs. We present a novel solution, called *TESLA*. Overall, *TESLA* in the field of LDP is somewhat similar to the role of homomorphic encryption in secure multi-party computation. In *TESLA*, to mitigate the negative influence of noise on truth aggregation, we design *RFM* and *PFM* to obtain the filtered values. To mitigate the negative influence of noise on weight estimation, we design *PWM* to model the mixed error distribution. We provide the theoretical analysis of *TESLA*'s utility, privacy and complexity. While being proposed for truth discovery under LDP, its idea is also applicable to other crowdsensing tasks (e.g. machine learning) while guaranteeing rigorous LDP.

CRediT authorship contribution statement

Pengfei Zhang: Conceptualization, Methodology, Data curation, Writing – original draft. **Xiang Cheng:** Project administration, Writing – review & editing, Funding acquisition. **Sen Su:** Writing – review & editing, Supervision, Funding acquisition. **Ning Wang:** Software, Validation.

Declaration of competing interest

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

Data availability

The authors do not have permission to share data.

Acknowledgments

The authors thank the reviewers for their valuable comments which significantly improved this paper. This work is the results of the research project funded by the National Natural Science Foundation of China [Grant No. 62072052 and No. 61872045] and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China [Grant No. 61921003].

References

[1] Chengmei Lv, Taochun Wang, Chengtian Wang, Fulong Chen, Chuanxin Zhao, ESPPTD: an efficient slicing-based privacy-preserving truth discovery in mobile crowd sensing, Knowl.-Based Syst. 229 (2021) 107–149.

[2] Shadan Ghaffaripour, Ali Miri, A decentralized, privacy-preserving and crowdsourcing-based approach to medical research, in: IEEE Conf. on Syst. Man, and Cybernetics, SMC, 2020, pp. 4510–4515.

[3] Mucheol Kim, Brij B. Gupta, Seungmin Rho, Crowdsourcing based scientific issue tracking with topic analysis, Appl. Soft Comput. 66 (2018) 506–511.

[4] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, Jiawei Han, A confidence-aware approach for truth discovery on long-tail data, PVLDB 8 (4) (2014) 425–436.

[5] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, Adam D. Smith, What can we learn privately? in: IEEE Symposium on Foundations of Comp. Sci., FOCS, 2008, pp. 531–540.

[6] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, Ge Yu, Collecting and analyzing multidimensional data with local differential privacy, in: IEEE Conf. on Data Engin., ICDE, 2019, pp. 638–649.

[7] Haipei Sun, Boxiang Dong, Wendy.Hui Wang, Ting Yu, Zhan. Qin, Truth inference on sparse crowdsourcing data with local differential privacy, IEEE Conf. on Big Data, Big Data (2018) 488–497.

[8] Yaliang Li, Chenglin Miao, Lu Su, Jing Gao, Qi Li, Bolin Ding, Zhan Qin, Kui. Ren, An efficient two-layer mechanism for privacypreserving truth discovery, in: Proc. of ACM Conf. on Knowl. Discovery and Data Mining, KDD, 2018, pp. 1705–1714.

[9] Yaliang Li, Houping Xiao, Zhan Qin, Chenglin Miao, Lu Su, Jing Gao, Kui Ren, Bolin. Ding, Towards differentially private truth discovery for crowd sensing systems, in: IEEE Conf. on Distributed Comput. Syst., ICDCS, 2020, pp. 1156–1166.

[10] Peng Sun, Zhibo Wang, Yunhe Feng, Liantao Wu, Yanjun Li, Hairong Qi, Zhi Wang, Towards personalized privacy-preserving incentive for truth discovery in crowdsourced binary-choice question answering, in: IEEE Conf. on Comp. Communi., INFOCOM, 2020, pp. 1133–1142.

[11] P Sun, Z Wang, L Wu, Y Feng, X Pang, H Qi, Z. Wang, Towards personalized privacy-preserving incentive for truth discovery in mobile crowdsensing systems, IEEE Trans. Mob. Comput. (2020) 1.

[12] Di Wang, Jinhui Xu, Inferring ground truth from crowdsourced data under local attribute differential privacy, Theoret. Comput. Sci. 865 (2021) 85–98.

[13] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, Jiawei Han, Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation, in: Proc. of ACM Conf. on Management of Data, SIGMOD, 2014, pp. 1187–1198.

[14] Houping Xiao, Jing Gao, Qi Li, Fenglong Ma, Lu Su, Yunlong Feng, Aidong Zhang, Towards confidence interval estimation in truth discovery, IEEE Trans. Knowl. Data Eng. 31 (3) (2019) 575–588.

[15] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, Kui Ren, Cloud-enabled privacypreserving truth discovery in crowd sensing systems, in: Proc. of ACM Conf. on Embedded Netw. Sensor Syst. SenSys, 2015, pp. 183–196.

[16] Yaqing Wang, Fenglong Ma, Lu Su, Jing. Gao, Discovering truths from distributed data, in: Proc. of IEEE Conf. on Data Mining, ICDM, 2017, pp. 505–514.

[17] Chen Ye, Hongzhi Wang, Tingting Ma, Jing Gao, Hengtong Zhang, Jianzhong Li, Patternfinder: Pattern discovery for truth discovery, Knowl.-Based Syst. 176 (2019) 97–109.

[18] Hengtong Zhang, Yaliang Li, Fenglong Ma, Jing Gao, Lu Su, Texttruth: An unsupervised approach to discover trustworthy information from multi-sourced text data, in: Proc. of ACM Conf. on Knowl. Discovery and Data Mining, KDD, 2018, pp. 2729–2737.

[19] Liuyi Yao, Lu Su, Qi Li, Yaliang Li, Fenglong Ma, Jing Gao, Aidong Zhang, Online truth discovery on time series data, in: Proc. of SIAM Conf. on Data Mining, SDM, 2018, pp. 162–170.

[20] John C. Duchi, Michael I. Jordan, Martin J. Wainwright, Local privacy and statistical minimax rates, in: Proc. of IEEE Sympo. on Founda. of Comput. Sci., FOCS, 2013, pp. 429–438.

[21] S.L. Warner, Randomized response: A survey technique for eliminating evasive answer bias, J. Amer. Statist. Assoc. 60 (309) (1965) 63–69.

[22] Peter Kairouz, Sewoong Oh, Pramod Viswanath, Extremal mechanisms for local differential privacy, in: Advances in Neural Information Processing Systems, NIPS, 2014, pp. 2879–2887.

[23] Úlfar Erlingsson, Vasyl Pihur, Aleksandra Korolova, RAPPOR: randomized aggregatable privacy-preserving ordinal response, in: Proc. of ACM SIGSAC Conf. on Comput and Commun. Secur., CCS, 2014, pp. 1054–1067.

[24] Tianhao Wang, Ninghui Li, Somesh Jha, Locally differentially private frequent itemset mining, in: IEEE Symposium on Secur and Priv., SP, 2018, pp. 127–143.

[25] Qingqing Ye, Haibo Hu, Xiaofeng Meng, Huadi Zheng, Privkv: Key–value data collection with local differential privacy, in: IEEE Symposium on Secur and Priv., SP, 2019, pp. 317–331.

[26] Lin Sun, Xiaojun Ye, Jun Zhao, Chenhui Lu, Mengmeng Yang, Bisample: Bidirectional sampling for handling missing data with local differential privacy, Database Syst. for Advan. Appli. (2020) 88–104.

[27] X Li, H Yan, G Zheng, et al., Key-value data collection with distribution estimation under local differential privacy, Secur and Communi. Netw. 2022 (1) (2022) 1–20.

[28] Cynthia Dwork, Frank McSherry, Kobbi Nissim, Adam D. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography, TCC, 2006, pp. 265–284.

[29] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, Kui Ren, Privacy-preserving truth discovery in crowd sensing systems, ACM Trans. Sens. Netw. 15 (2019) 32, 9:1–9.

[30] Yifeng Zheng, Huayi Duan, Xingliang Yuan, Cong Wang, Privacy-aware and efficient mobile crowdsensing with truth discovery, IEEE Trans. Depend. Secur. Comput. 17 (1) (2020) 121–133.

[31] Chenglin Miao, Lu Su, Wenjun Jiang, Yaliang Li, Miaomiao. Tian, A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems. In, in: Proc. of IEEE Conf. on Comput. Commun., INFOCOM, 2017, pp. 1–9.

[32] Xiaoting Tang, Cong Wang, Xingliang Yuan, Qian Wang, Non-interactive privacy-preserving truth discovery in crowd sensing applications, in: Proc. of IEEE Conf. on Comput. Commun., INFOCOM, 2018, pp. 1988–1996.

[33] Guowen Xu, Hongwei Li, Sen Liu, Mi Wen, Rongxing Lu, Efficient and privacy-preserving truth discovery in mobile crowd sensing systems, IEEE Trans. Veh. Technol. 68 (4) (2019) 3854–3865.

[34] Chuan Zhang, Liehuang Zhu, Chang Xu, Kashif Sharif, Xiaojiang Du, Mohsen Guizani, LPTD: achieving lightweight and privacypreserving truth discovery in ciot, Future Gener. Comput. Syst. 90 (2019) 175–184.

[35] Yifeng Zheng, Huayi Duan, Cong Wang, Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing, IEEE Trans. Inf. Forensics Secur. 13 (10) (2018) 2475–2489.

[36] Jianchao Tang, Shaojing Fu, Ming Xu, Yuchuan Luo, Kai Huang, Achieve privacy-preserving truth discovery in crowdsensing systems, in: Proc. of ACM Conf. on Inf and Knowl. Management, CIKM, 2019, pp. 1301–1310.

[37] Jingsheng Gao, Shaojing Fu, Yuchuan Luo, Tao Xie, Location privacy-preserving truth discovery in mobile crowd sensing, in: Proc. of Conf. on Comput. Commun and Netw., ICCCN, 2020, pp. 1–9.

[38] Yuxian Liu, Shaohua Tang, Hao-Tian Wu, Xinglin Zhang, RTPT: A framework for real-time privacy-preserving truth discovery on crowdsensed data streams, Comput. Netw. 148 (2019) 349–360.

[39] Guowen Xu, Hongwei Li, Shengmin Xu, Hao Ren, Yinghui Zhang, Jianfei Sun, Robert H. Deng, Catch you if you deceive me: Verifiable and privacy-aware truth discovery in crowdsensing systems, in: Proc. of ACM Asia Conf. on Comput and Commun. Secur., AsiaCCS, 2020, pp. 178–192.

[40] Zongda Wu, Shigen Shen, Huxiong Li, Haiping Zhou, Chenglang Lu, A basic framework for privacy protection in personalized information retrieval: An effective framework for user privacy protection, J. Organ. User Comput. 33 (6) (2021) 1–26.

[41] Zongda Wu, Guiling Li, Shigen Shen, Xinze Lian, Enhong Chen, Guandong Xu, Constructing dummy query sequences to protect location privacy and query privacy in location-based services, World Wide Web 24 (1) (2021) 25–49.

[42] Zongda Wu, Shigen Shen, Haiping Zhou, Huxiong Li, Chenglang Lu, Dongdong Zou, An effective approach for the protection of user commodity viewing privacy in e-commerce website, Knowl.-Based Syst. 220 (2021) 106952.

[43] Z Wu, R Wang, Q Li, et al., A location privacy-preserving system based on query range cover-up or location-based services, IEEE Trans. Vehi. Technol. 69 (5) (2020) 5244–5254.

[44] Zongda Wu, Shigen Shen, Xinze Lian, Xinning Su, Enhong Chen, A dummy-based user privacy protection approach for text information retrieval, Knowl.-Based Syst. 195 (2020) 105679.

[45] Zongda Wu, Renchao Li, Zhifeng Zhou, Junfang Guo, Jionghui Jiang, Xinning Su, A user sensitive subject protection approach for book search service, J. Assoc. Inf. Sci. Technol. 71 (2) (2020) 183–195.

[46] Zongda Wu, Guiling Li, Qi Liu, Guandong Xu, Enhong Chen, Covering the sensitive subjects to protect personal privacy in personalized recommendation, IEEE Trans. Serv. Comput. 11 (3) (2018) 493–506.

[47] Zongda Wu, Guandong Xu, Chenglang Lu, Enhong Chen, Fang Jiang, Guiling Li, An effective approach for the protection of privacy text data in the CloudDB, World Wide Web 21 (4) (2018) 915–938.

[48] Zongda Wu, Jie Shi, Chenglang Lu, Enhong Chen, Guandong Xu, Guiling Li, Sihong Xie, Philip S. Yu, Constructing plausible innocuous pseudo queries to protect user query intention, Inform. Sci. 325 (2015) 215–226.

[49] Tianyi Li, Yu Gu, Xiangmin Zhou, Qian Ma, Ge Yu, An effective and efficient truth discovery framework over data streams, 2017, pp. 180–191, EDBT.

[50] Alexandre B. Tsybakov, Introduction to Nonparametric Estimation, in: Springer series in statistics, Springer, 2009.

[51] Thông T. Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, Junbum Shin, Collecting and analyzing data from smart device users with local differential privacy, 2016, CoRR abs/1606.05053.

Pengfei Zhang is a Ph.D. Candidate from Beijing University of Posts and Telecommunications in China. His major is Computer Science. His current research interest focuses on privacy protection in mobile crowdsensing systems.

Xiang Cheng received the Ph.D. Degree in Computer Science from Beijing University of Posts and Telecommunications, China, in 2013. He is currently a Professor at the Beijing University of Posts and Telecommunications. His research interests include privacy-enhanced computing, data mining and knowledge engineering.

Sen Su received the Ph.D. degree in 1998 from the University of Electronic Science and Technology, Chengdu, China. He is currently a Professor at the Beijing University of Posts and Telecommunications. His research interests include data privacy, cloud computing and internet services.

Ning Wang is a graduate student from Beijing University of Posts and Telecommunications in China. His major is Computer Science. His research interests include data mining and data privacy.