ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230420

Integrated Private Data Trading Systems for Data Marketplaces

Weidong Li[a,b;*], **Mengxiao Zhang**[a,c;**], **Libo Zhang**[b] **and Jiamou Liu**[b]

[a]School of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[b]School of Computer Science, The University of Auckland, New Zealand
[c]Department of Computer Science, Durham University, U.K.

Abstract. In the digital age, data is a valuable commodity, and data marketplaces offer lucrative opportunities for data owners to monetize their private data. However, data privacy is a significant concern, and differential privacy has become a popular solution to address this issue. Private data trading systems (PDQS) facilitate the trade of private data by determining which data owners to purchase data from, the amount of privacy purchased, and providing specific aggregation statistics while protecting the privacy of data owners. However, existing PDQS with separated procurement and query processes are prone to over-perturbation of private data and lack trustworthiness. To address this issue, this paper proposes a framework for PDQS with an integrated procurement and query process to avoid excessive perturbation of private data. We also present two instances of this framework, one based on a greedy approach and another based on a neural network. Our experimental results show that both of our mechanisms outperformed the separately conducted procurement and query mechanism under the same budget regarding accuracy.

1 Introduction

In the modern digital era, data have become a highly valuable asset, providing opportunities to gain insights, make informed decisions, and drive innovation across various domains, leading to significant economic benefits for individuals, businesses, and governments [26]. However, this abundance of data also raises a pressing concern: data privacy leakage. With recent data breaches affecting millions of 533 Facebook users[1], 150,000 NHS patients[2] and fitness tracking app Strava users[3], privacy preservation has become increasingly critical. This creates an inherent conflict between preserving privacy for individuals and unlocking significant economic value for society. *Data marketplaces* have emerged as a potential solution to this issue, as they compensate *data owners* for their private data to offset the potential privacy loss. By providing compensation, data owners may be more willing to share their data, allowing it to be further utilised [37].

Imagine a scenario where a data analyst, also referred to as a *data consumer*, wishes to obtain aggregation statistics from individual data owners through a data marketplace. These data owners are willing to share their private data on the data marketplace as long as they receive fair compensation. Each data owner has a specific compensation amount in mind, known as their *valuation*, which they keep to themselves. Additionally, the data owners require their privacy to be safeguarded to a certain extent. To facilitate this transaction, a *data broker* acts as an intermediary between the data owners and data consumers on behalf of the data marketplace, procuring data from data owners and analysing the collected data for data consumers.

A *private data trading system* (PDQS) [40] is a mechanism that enables the trade of private data, determining how much to compensate the data owners, generating specific aggregation statistics, and ensuring the privacy protection of the data owners. Starting with Ghosh and Roth's work [20], a typical PDQS consists of two main components: a *procurement process* determines the selection of data owners from whom data will be purchased and the payment for the selected data owners; a *query process* executes a query on the procured data, adds noise to the data, and outputs query results.

Differential privacy (DP) [15] is a privacy concept that can be employed to measure the degree of privacy loss for each data owner, indicating the amount of privacy compromised during the data trading. A typical method to achieve DP is to execute the query on the *raw* datasets and add noise to the true query answer. In this case, the data broker, who executes the query, is deemed to be trustworthy and has the full access to the raw data. However, in practice, the data broker may not be trustworthy: once obtaining the raw data, the data broker may resell the data to third parties for profit. Given that, a local model of DP, known as *local differential privacy* (LDP) [14] is proposed. In LDP model, each data owner adds noise to her private data locally before sharing it to the data broker. Most of the existing PDQS deploys the DP model [20, 13, 11, 39]. We instead extend the PDQS design to the LDP model.

Existing works for implementing PDQS execute the procurement and the query processes one after another [17]. Specifically, the procurement process first selects a subset of data owners. Then to achieve LDP, the query process runs a *local randomiser* [16] where a smaller subset of data owners are selected randomly and submit their true data while each of the others submit a random number. In this way, the data owners who submit random numbers are paid with no contribution to the query in the query process. In other words, the budget is not put to good use and the query accuracy could be improved. Therefore, a question arises: *How to design a PDQS with an integrated procurement and query process to address this issue?*

A challenge arises when designing such an integrated procurement and query process. The procurement process and the query process

* Corresponding Author. Email: wli916@aucklanduni.ac.nz
** Corresponding Author. Email: mengxiao.zhang@uestc.edu.cn
[1] https://www.bbc.com/news/technology-56815478
[2] https://www.bbc.com/news/technology-44682369
[3] https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

serve for different purposes and thus are expected to have different properties: the procurement process is used to incentivise the data owners to reveal their valuations for pricing purposes and it should satisfy incentive compatibility (IC), individual rationality (IR) and budget feasibility (BF) properties (see formal definitions in Sec. 3), while the query process should satisfy LDP. Therefore, we need to design proper rules of the integrated process that meets all of these properties and thus can serve well for both procurement and query processes.

To address the research question, we propose a new framework of the PDQS, named *Integrated PDQS*. The core of this framework is embedding the procurement process into the query process such that data owner selection and data perturbation happen simultaneously. To be specific, in the new framework, we first assign an allocation probability that a data owner would be procured, then we use this probability to obfuscate the data in the local randomiser. In this way, all selected data owners, once being paid, contribute to the query answer. See details in Sec. 4.

Then we propose two PDQS instantiations that implement the proposed framework. The first mechanism, **g**reedy **p**rivate **q**uery **m**echanism (GPQM), selects data owners with lower valuation in a greedy manner until the budget is exhausted. This mechanism allows us to use any non-increasing function as the allocation probability, but requires manual selection of the function and its parameters. Our second mechanism, **n**eural-based **p**rivate **q**uery **m**echanism (NPQM), parameterises the allocation probability with a neural network. The parameters are learned with the dual-ascent algorithm. We theoretically proved that both GPQM and NPQM satisfy desirable properties. See details in Sec. 5 and 6. We also empirically validate the performance of the propose two PDQS instances. Experimental results show that both of our proposed methods outperform the benchmarks, with NPQM performing better than GPQM.

The contributions of this paper are summarised as follows:

- We propose a new framework of PDQS that integrates the procurement and query process.
- We design two PDQS instances to implement the propose framework: GPQM is equipped with a non-decreasing function as the allocation probability, and NPQM uses neural network to learn an optimal allocation probability.
- We empirically show the strengths of the two proposed PDQSs.

2 Related works

In their seminal work, Ghosh and Roth [20] formalised the concept of trading private data under differential privacy with their proposed mechanism, FairQuery, which includes independent procurement and query processes. In the procurement process, a subset of data owners is selected based on their bids, and their private data is reported to the system without perturbation. In the query process, random noise is added to the query answer based on the collected dataset to guarantee DP. This leads to FairQuery and its extensions e.g. FairInnerProduct [13] heavily relying on a trusted data broker.

To tackle the distrust between data owners and the data broker, Wang et al. [34] and Fallah et al. [17] proposed PDQSs by enabling data owners to perturb their private data locally to ensure local DP. However, they do not consider the budget constraints of the data consumer. Additionally, they rely on separated procurement and query processes, leading to excessive perturbation of private data.

There are other works that extended Ghosh and Roth's work from different perspectives, e.g., considering the correlations between data owners' values and valuations [33, 18, 24, 30, 9, 10], the cases that data owners' private data is not verifiable [19], the scenario where different levels of data accuracy are provided by various data brokers [12], the network effect on data owners' participation [41, 22], single-minded data owners [38], data owners getting benefits from the statistic based on reported private data [11, 17]. There are also studies working on the design of truthful mechanisms for trading data without preserving data under differential privacy [7, 25], privacy-aware mechanisms that preserve user bids [28, 29, 1, 23], pricing mechanisms that charging users based on their perturbed private data [8], and pricing strategies based on data quality [36].

To the best of our knowledge, there is no existing PDQS addressing our problem of trading private data under LDP while integrating procurement and query processes and ensuring IC, IR and BF properties (formally defined in Sec. 3). Thus, we propose our framework to solve this problem and two instantiations of the framework.

3 Problem formulation

Consider a data transaction with a data consumer and n data owners. The data consumer aims to obtain some aggregation information about the data owners, e.g., how many people are infected in the population, which can be denoted as a *query* f. The data consumer has a budget, denoted by $\beta \in \mathbb{R}_+$, for the query.

Each data owner i has private data $t_i \in \{0, 1\}$. The data owner is willing to sell her private data to the data consumer, given reasonable compensation and privacy protection. Let $\varepsilon_i \in \mathbb{R}_+$ be a *privacy parameter*. When her data is used in an ε_i-differential privacy manner, she suffers a privacy cost $c_i := \varepsilon_i \theta_i$, where $\theta_i \in [\underline{\theta}, \overline{\theta}]$ is her *valuation* to a unit of privacy. A data owner can be represented by a tuple $s_i := (t_i, \theta_i)$. We use $\vec{s} := (s_1, \ldots, s_n)$ to denote the data owners, $\vec{t} := (t_1, \ldots, t_n)$ and $\vec{\theta} := (\theta_1, \ldots, \theta_n)$ to denote the private data and the valuation vector of all n data owners, respectively. Also, we use $\vec{\theta}_{-i} := (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n)$ to denote the valuation vector of all data owners but s_i. Data owners may misreport their valuations to, for example, gain more compensation. Let $b_i \in [\underline{\theta}, \overline{\theta}]$ denote the reported valuation of data owner s_i. Similarly, we have vectors $\vec{b} := (b_1, \ldots, b_n)$ and $\vec{b}_{-i} := (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$.

We are committed to developing a Private Data Query System (PDQS) that takes the set of data owners and the data consumer's budget as inputs. The PDQS consists of four components: an *allocation function* that determines the selection probabilities of data owners, a *payment function* that calculates the compensation for each data owner, an *LDP algorithm* that ensures privacy by adding noise locally, and a *query function* that implements the query f based on the collected data. Through the collaboration of these components, the PDQS efficiently processes queries while ensuring the privacy of data owners.

Next, we formally define these functions and outline the desired properties they should possess.

Definition 1 *An* allocation function $q : [\underline{\theta}, \overline{\theta}]^n \to [0, 1]^n$ *is a mapping from the bids of all data owners to an allocation result* $\vec{q} := (q_1, \ldots, q_n)$*, where* q_i *is the probability that data owner* s_i *is selected.*

We also write $q_i(\vec{b})$ as the allocation function of an individual data owner s_i. When the context is clear, we write q_i for short.

Definition 2 *A* payment function $p : [\underline{\theta}, \overline{\theta}]^n \to {\mathbb{R}_+}^n$ *is a mapping from the bids of all data owners, represented by payment vector* $\vec{p} := (p_1, \ldots, p_n)$*, where* p_i *is the compensation of data owner* s_i*.*

Note that when the data owners make their decisions on bidding, they have no idea about whether they would be selected or not. We assume that the data owners are rational and make decisions based on the expected benefit.

Let $P_i = p_i q_i$ be the expected payment of s_i. The expected utility of a data owner s_i is the difference between her expected compensation and the valuation of expected privacy loss, i.e., $u_i := P_i - \varepsilon_i \theta_i q_i$.

The allocation function together with the payment function is expected to satisfy the following properties.

- **Incentive compatibility (IC).** Each data owner $s_i, 1 \leq i \leq n$ maximises her expected utility when reporting true valuation, i.e.,

$$u_i(\theta_i, \vec{b}_{-i}) \geq u_i(b_i, \vec{b}_{-i}) \, \forall b_i \neq \theta_i, \forall \vec{b}_{-i} \in [\underline{\theta}, \overline{\theta}]^{n-1}. \quad (1)$$

- **Individual rationality (IR).** Each data owner $s_i, 1 \leq i \leq n$ gets non-negative expected utility when reporting true valuation, i.e.,

$$u_i(\theta_i, \vec{b}_{-i}) \geq 0 \, \forall \vec{b}_{-i} \in [\underline{\theta}, \overline{\theta}]^{n-1}. \quad (2)$$

- **Budget feasibility (BF).** The total expected payment is within the budget β specified by the data consumer, i.e.,

$$\sum_{i=1}^{n} p_i(\vec{b}) q_i(\vec{b}) \leq \beta. \quad (3)$$

Intuitively, IC ensures the data owners are incentivised to report their true valuations as doing so leads to the best utility while IR ensures that the data owners are willing to participate in the system as it leads to at least non-negative utility.

Definition 3 *A* query $f : \mathbb{R}^n \to \mathbb{R}$ *is a mapping from private data of* n *data owners to a real value.*

Commonly used queries include count and median queries. Count query counts the number of 1s in a binary dataset, i.e., $f(\vec{t}) = \sum_{i=1}^{n} t_i$, while median query returns the median number in a real-valued dataset.

Considering the privacy concern, we use local differential privacy (LDP) [14] as the privacy concept. In order to ensure LDP, a *local randomiser* is commonly utilised, allowing data owners to obfuscate their private data before sharing it for answering queries.

Let $\mathcal{D}$ denote the domain of private data and Ω be a probability space. We define the local randomiser and LDP as follows.

Definition 4 *A* local randomiser $\mathcal{L} : \mathcal{D} \times \Omega \to \mathcal{D}$ *is a randomised function mapping a private value* t_i *to a random value* $t_i' \in \mathcal{D}$.

Definition 5 ([14]) *A local randomiser* $\mathcal{L}$ *is* ε_i-local differentially private*, if for any pair of input* $t_i, t_i' \in \mathcal{D}$ *and for any possible output* $o \in Range(\mathcal{L})$*, we have*

$$\Pr[\mathcal{L}(t_i) = o] \leq e^{\varepsilon_i} \Pr[\mathcal{L}(t_i') = o],$$

where ε_i is a non-negative real number to measure data owner s_i's privacy loss.

For a given query f and PDQS M, we use (α, δ)-probably approximately correct ((α, δ)-PAC) to measure M's accuracy.

Definition 6 *For* $\alpha, \delta \in [0, 1]$*, a private data query system* M *is* (α, δ)-probably approximately correct *((α, δ)-PAC) if for any dataset* $\vec{t} = (t_i, \ldots, t_n)$*,*

$$\Pr[|M(\vec{t}) - f(\vec{t})| \geq \alpha] \leq 1 - \delta,$$

where $|M(\vec{t}) - f(\vec{t})|$ is the difference between the query answer derived by M and the true answer.

The goal of this research is to develop a PDQS that meets IC, IR, BF, and ε_i-LDP properties, while also approximating query accuracy.

4 Proposed framework

We propose the framework *Integrated PDQS*, which combines the procurement and query processes into an integrated system. Integrated PDQS receives bids, private data, the budget, the query and an allocation function as inputs, and provides a payment result and a query answer as outputs. This framework can be instantiated to create PDQSs that satisfy the desired properties, i.e., IC, IR, BF and ε_i-LDP. The workflow of the framework is shown in Figure 1.

Unlike existing PDQSs that separate the procurement and query processes, the Integrated PDQS assigns probabilities to each data owner. These probabilities determine both the likelihood of being selected by the system and the likelihood of reporting their true private data. Consequently, the consumer's budget is utilised more efficiently, leading to higher query accuracy.[4]

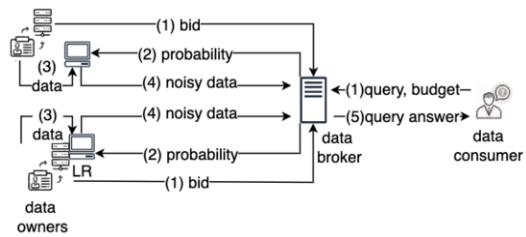

Figure 1: Workflow of proposed Integrated PDQS framework

Algorithm 1 outlines the specific steps of the Integrated PDQS framework. Initially, the framework calculates the allocation probabilities and expected payments for each data owner (Lines 1-2 of Alg. 1). Unlike existing approaches that select data owners before the query process, the Integrated PDQS framework incorporates an *Integrated Local Randomiser* (ILR) $\mathcal{L}_I$ for each data owner (Lines 3-5 of Alg. 1). As shown in Alg. 2, $\mathcal{L}_I$ is employed to select a data owner based on her probability q_i. If a data owner is chosen, her true private data is reported, and her compensation is calculated as P_i/q_i. On the other hand, if a data owner is not selected, a random value is reported, and no compensation is provided. Subsequently, the framework performs the query on the noisy dataset obtained from the data owners (Line 6 of Alg. 1).

In the framework, the procurement process is embedded in the query process through the use of ILR. In contrast to the traditional local randomiser, ILR not only determines the reported private data but also determines the compensation for each data owner. This design ensures that only the data owners who disclose their true data are selected and receive payment.

Algorithm 1 The Integrated PDQS framework

Input: Data owners $\vec{s}$, budget β, query f, allocation function q
Output: Payment vector $(p_1, \ldots, p_n)$, query answer z
1: Generate the allocation vector $(q_1, \ldots, q_n)$
2: Generate the payment vector $(P_1, \ldots, P_n)$
3: **for** each data owner s_i **do**
4: Get a random value and the payment $t_i', p_i = \mathcal{L}_I(t_i, q_i, P_i)$
5: **end for**
6: Compute the query answer $z = f(t_1', \ldots, t_n')$
7: Return $(p_1, \ldots, p_n), z$

Applying the ILR with probability q_i for data owner s_i ensures $\ln \frac{1+q_i}{1-q_i}$-local differential privacy, as shown in the following lemma.

[4] Note that the randomised approach, under the premise of privacy protection, achieves the goal of compensating only those owners who report true data, which deterministic procurement mechanisms cannot accomplish.

Algorithm 2 Ingerated Local Randomiser $\mathcal{L}_I$

Input: Private data t_i, probability q_i, expected payment P_i
Output: Perturbed data entry $\hat{t}_i$
1: With probability q_i, set $\hat{t}_i = t_i, p_i = P_i/q_i$; with probability $1-q_i$, set $\hat{t}_i$ to be a random value $t_i' \in \mathcal{D}, p_i = 0$
2: Return $\hat{t}_i, p_i$

Lemma 1 *The Integrated Local Randomiser $\mathcal{L}_I$ with probability q_i is ε_i-local differential privacy, where $\varepsilon_i = \ln \frac{1+q_i}{1-q_i}$.*

Proof. Given two data entries t_i, t_i', and $t_i \neq t_i'$, for $r \in \{0,1\}$: when $r = t_i$,

$$\frac{\Pr[\mathcal{L}_I(t_i) = r]}{\Pr[\mathcal{L}_I(t_i') = r]} = \frac{q_i + \frac{1}{2}(1-q_i)}{\frac{1}{2}(1-q_i)} = \frac{1+q_i}{1-q_i} \leq e^{\varepsilon_i};$$

when $r = t_i'$,

$$\frac{\Pr[\mathcal{L}_I(t_i) = r]}{\Pr[\mathcal{L}_I(t_i') = r]} = \frac{\frac{1}{2}(1-q_i)}{q_i + \frac{1}{2}(1-q_i)} = \frac{1-q_i}{1+q_i} \leq e^{\varepsilon_i}.$$

■

Note that when q_i approaches 1, the privacy guarantee $\varepsilon_i = \ln \frac{1+q_i}{1-q_i}$ is meaningless as it is infinity. Hence, from now on, we require the allocation probability $q_i \in [0,1)$.

Our goal is to design PDQSs that implement the Integrated PDQS framework, and satisfy IC, IR, BF and ε_i-LDP while approximating query accuracy. Any PDQS that instantiates the proposed framework employs ILR, so ε_i-LDP is guaranteed.

For query accuracy, we introduced the accuracy notion of (α, δ)-PAC in Section 3. To achieve high accuracy, we adopt the principle of Purchased Privacy Expectation Maximisation (PPEM) proposed in [38]. The PPEM principle highlights that acquiring a certain amount of expected privacy is a necessary condition for attaining a particular level of PAC accuracy. Thus, we may maximise the total expected purchased privacy to approximate the query accuracy. According to Lemma 1, the PDQS that adopts the proposed framework purchases the privacy of $\ln \frac{1+q_i}{1-q_i}$ with a probability q_i from data owner s_i. As a result, the total expected purchased privacy in the PDQS is given by $\sum_{i=1}^{n} q_i \ln \frac{1+q_i}{1-q_i}$.

Then, we characterise IC and IR properties. We define $w_i(b_i, b_{-i}) := q_i(b_i, b_{-i}) \ln \frac{1+q_i(b_i,b_{-i})}{1-q_i(b_i,b_{-i})}$ as the expected privacy loss of data owner s_i, when she bids b_i and others bid b_{-i}. For brevity, we refer to $w_i(b_i, b_{-i})$ as w_i when the context is clear. Applying Archer and Tardos's theorem [4] and considering w_i as the amount of *load* assigned to each data owner, we can derive the following theorem:

Theorem 2 ([4]) *A PDQS M satisfies IC and IR if and only if*

1. *the expected privacy loss $w_i(b_i, b_{-i})$ does not increase with respect to the bid b_i,*
2. *the expected privacy loss $w_i(b_i, b_{-i})$ satisfies $\int_0^{\overline{\theta}} w_i(x, b_{-i})dx < \infty$ for all i, b_{-i}, and*
3. *the expected payment is in the form of $P_i = b_i w_i(b_i, b_{-i}) + \int_{b_i}^{\overline{\theta}} w_i(x, b_{-i})dx$.*

Therefore we construct the optimisation problem that maximises the total expected purchased privacy while satisfying the IC, IR and BF constraints.

As the privacy loss $\varepsilon_i = \ln \frac{1+q_i}{1-q_i}$ of data owner s_i when she is selected with probability q_i increases with respect to q_i in the domain of $[0,1)$, we can observe that Condition 1 of Theorem 2 is satisfied if the allocation function q is non-increasing. Therefore, in the subsequent discussion, we impose the requirement of a non-increasing allocation function to fulfil Condition 1.

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} q_i \ln \frac{1+q_i}{1-q_i} \\ \text{s.t.} \quad & q_i(b_i', b_{-i}) \leq q_i(b_i, b_{-i}), \forall b_i' > b_i \\ & \sum_{i=1}^{n} [b_i w_i + \int_{b_i}^{\overline{\theta}} w_i(x, b_{-i})dx] \leq \beta \\ & 0 \leq q_i < 1, \forall i \in [0,n] \end{aligned} \tag{4}$$

Now the problem is to find the proper allocation function q to solve (4). In the following sections, we propose two instances of the proposed framework to solve the problem.

5 Greedy Private Query Mechanism (GPQM)

The core of designing an instance to implement the Integrated PDQS framework is the design of an allocation function q that addresses 4. A straightforward idea is to deploy a non-increasing function as the allocation function such that a data owner with a high valuation has a low chance to be selected, and then, according to the distribution, greedily choose data owners until the budget is used up. Such non-increasing allocation function can be a linear function e.g. $q_i = 1 - b_i$, a logarithmic function e.g. $q_i = -\log(b_i)$, or an exponential function e.g. $q_i = e^{-b_i}$. We refer to the greedy instance as **g***reedy* **p***rivate* **q***uery* **m***echanism* (GPQM).

To be specific, given a non-increasing allocation function, GPQM first determines the allocation probability q_i. Also, for each data owner s_i whose expected privacy loss is $w_i = q_i \ln \frac{1+q_i}{1-q_i}$, set her expected payment as $P_i = b_i w_i + \int_{b_i}^{\overline{\theta}} w_i(x, b_{-i})dx$. Then GPQM sorts the data owners by their allocation probabilities in descending order. Following the order, each data owner runs the ILR $\mathcal{L}_I$ until the budget is used up. GPQM finally returns the payment vector and the query answer. See Algorithm 3.

Algorithm 3 Greedy Private Query Mechanism (GPQM)

Input: Data owners $\vec{s}$, budget β, query f, non-increasing allocation function q
Output: Payment vector $(p_1, \ldots, p_n)$, query answer z
1: Generate the allocation vector $(q_1, \ldots, q_n) = q(\vec{b})$
2: Compute the expected payment vector $(P_1, \ldots, P_n)$, where $P_i = b_i w_i + \int_{b_i}^{\overline{\theta}} w_i(x, b_{-i})dx$
3: Sort the data owners in descending order with respect to q_i
4: Initialise $k = 1$
5: **while** $\sum_{i=1}^{k} P_i \leq \beta$ **do**
6: Get a random value and the payment $t_i', p_i = \mathcal{L}_I(t_i, q_i, P_i)$
7: Increment $k = k + 1$
8: **end while**
9: Set $p_i = 0$, if $i > k$
10: Compute the query answer $z = f(t_1', \ldots, t_n')$
11: Return $(p_1, \ldots, p_n), z$

Lemma 3 *Greedy private query mechanism (GPQM) is IC and IR.*

Proof. We prove the lemma by Archer and Tardos' theorem [4]. As GPQM employs a non-increasing allocation function, i.e., the allocation q_i does not increase with respect to the bid b_i, and the expected payment P_i is in the form specified in Line 2 of Alg. 3, Conditions 1 and 3 are met.

Now we show that Condition 2 is also met. We rewrite the integral as $\int_0^{\overline{\theta}} q_i(x)\ln(1+q_i(x)) - q_i(x)\ln(1-q_i(x))dx$, where $q_i(x)$ is short of $q_i(x, b_{-i})$. The former part is finite, so we focus on the latter part. We have

$$\int_0^{\overline{\theta}} -q(x)\ln(1-q(x))dx \le \int_0^{\overline{\theta}} -\ln(1-q(x))dx$$
$$\le -x\ln(1-\overline{q}) \le -\overline{\theta}\ln(1-\overline{q}),$$

where $\overline{q}$ is the maximum value of $q(x)$ in its domain. As the range of q is $[0,1)$, we have $\overline{q} = 1-\tau$, where τ is a very small positive real. We then have $-\ln(1-\overline{q}) \le -\ln\tau$, which is finite.

■

Theorem 4 *The greedy private query mechanism (GPQM) is IC, IR, BF and LDP.*

The mechanism is BF by construction. As the ILR $\mathcal{L}_I$ is applied to perturb the private data, the mechanism satisfies LDP.

6 The neural-based private query mechanism (NPQM)

We introduce another instance that implements the proposed framework. Unlike GPQM, this instance addresses (4) by learning the allocation function that approximates the total purchased privacy while satisfying the given constraints. To achieve this, we apply QMIX [32] and neural-network techniques to design and parameterise the allocation function, and learn its parameters using dual ascent algorithm [6]. We refer to this instance as **n***eural-based* **p***rivate* **q***uery* **m***echanism* (NPQM).

6.1 Allocation function design and parametrisation

The allocation function in NPQM is designed as $q_i(\vec{b}) = \sigma(|w_2|(-|w_1|b_i + c) + d)$, where σ denotes the sigmoid function, i.e., $\sigma(x) = \frac{e^x}{e^x+1}$, and w_1, w_2, c, d are parameters.

The parameters w_1, w_2, c, d are generated by separate neural networks. Specifically, we use three separate three-layer neural networks to generate w_1, w_2, and d, respectively. The neural network is composed of three functions, which are:

- A linear function $l^{(1)}: [\overline{\theta}, \underline{\theta}]^n \to \mathbb{R}^h$ takes the bids as input. It has the form $l^{(1)}(\vec{b}) = A^{(1)}\vec{b} + k^{(1)}$ where $A^{(1)}$ and $k^{(1)}$ are coefficients of the linear function, and h is a constant denoting the number of neural employed in Layer 2;
- A function $l^{(2)}: \mathbb{R}^h \to \mathbb{R}^h$ takes the result from Layer 1. In detail, $l^{(2)}(x) = x$ if $x \ge 0$; $l^{(2)}(x) = 0$ otherwise. This is known as a ReLU function;
- A linear function $l^{(3)}: \mathbb{R}^h \to \mathbb{R}$ takes the the result from Layer 2. It has the form $l^{(3)}(\vec{b}) = A^{(3)}\vec{b} + k^{(3)}$ where $A^{(3)}$ and $k^{(3)}$ are coefficients of the linear function.

The parameters w_1, w_2, d are computed by $l^{(3)}(l^{(2)}(l^{(1)}(\vec{b})))$, which forms a function from $[\overline{\theta}, \underline{\theta}]^n$ to $\mathbb{R}$. Notice that although three parameters w_1, w_2, and d are calculated by the same form of function, they have different values since the coefficients in $l^{(3)}$ and $l^{(1)}$ are different, e.g. $A^{(1)}$ for w_1 and w_2 can be different.

The parameter c is directly computed by a single linear function $l^{(c)}: [\overline{\theta}, \underline{\theta}]^n \to \mathbb{R}$ with the form $l^{(c)}(\vec{b}) = A^{(c)}\vec{b} + k^{(c)}$. Let $\mu = (w_1, w_2, c, d)$ denote all learnable parameters in the parameterised model, then we denote the parameterised allocation function as q^μ.

This type of function and neural network design has been previously utilised in QMIX [32], a well-known deep multi-agent reinforcement learning algorithm, which ensures the monotonicity between the global and the local value function. The details of QMIX is described in Appendix B of the full paper.

The optimal values of the coefficients in each layer will be trained and approximated by the Stochastic Gradient Descent (SGD) method. The performance of the neural network thus can be guaranteed by the proper settings of coefficients in each layer.

6.2 Learning the parameters of the allocation function

We employ dual ascent techniques [6] to learn the parameter μ specified above and approximate the optimal solution to (4). We begin with rewriting (4) with μ as the following.

$$\max_{\mu} \sum_{i=1}^{n} q_i^\mu \ln\frac{1+q_i^\mu}{1-q_i^\mu} \tag{5}$$

$$\text{s.t. } q_i^\mu(b_i', b_{-i}) \le q_i^\mu(b_i, b_{-i}), \forall b_i' > b_i \tag{6}$$

$$\sum_{i=1}^{n}\left(b_i q_i^\mu \ln\frac{1+q_i^\mu}{1-q_i^\mu} + \int_{b_i}^{\overline{\theta}} q^\mu(x)\ln\frac{1+q^\mu(x)}{1-q^\mu(x)}dx\right) - \beta \le 0 \tag{7}$$

$$0 \le q_i^\mu < 1, \forall i \in [0, n] \tag{8}$$

Recall that n is the number of data owners, and q_i is the probability that data owner s_i is selected by the system. The objective function aims to maximise the total expected privacy purchased by the system, where $\ln\frac{1+q_i^\mu}{1-q_i^\mu}$ is the privacy loss ε_i of data owner s_i, if she is selected Constraint (6) ensures that, for any data owner s_i, the allocation function is non-increasing with respect to her bid b_i, given other data owners' bids are fixed. Constraint (7) ensures that the total expected payment does not exceed the given budget. Constraint (8) ensures that the probability of a data owner being selected is less than 1, thereby providing privacy protection for each data owner.

Constraint (6) is satisfied since the parameterised model is enforced to be a monotonic function. Constraint (8) is also satisfied as we used the Sigmoid function, and the output is constrained between 0 and 1. Thus, we omit these two constraints in the following formulations.

Next, we discuss how to update the parameters of the parameterised allocation function q^μ to satisfy the BF constraint (as shown in (7)) and approximate the total purchased privacy (as shown in (5)) by the dual ascent algorithm [6].

We establish the dual problem of the primal problem (5). Let $\phi(\mu) = \sum_{i=1}^n q_i^\mu \ln\frac{1+q_i^\mu}{1-q_i^\mu}$ denote the objective function of the primal problem, representing the total expected purchased privacy of the system. Let $g(\mu) = \sum_{i=1}^n \left(b_i q_i^\mu \ln\frac{1+q_i^\mu}{1-q_i^\mu} + \int_{b_i}^{\overline{\theta}} q^\mu(x)\ln\frac{1+q^\mu(x)}{1-q^\mu(x)}dx\right) - \beta$ denote the difference between the total expected payment and the budget. The standard form of the primal problem is

$$\min_{\mu} -\phi(\mu) \tag{9}$$

$$\text{s.t. } g(\mu) \le 0 \tag{10}$$

The Lagrangian is $L(\mu, \lambda) = -\phi(\mu) + \lambda g(\mu)$, where $\lambda \ge 0$ is the Lagrangian multiplier. The dual function of the optimisation problem defined by (9), (10) is $\psi(\mu, \lambda) = \inf L(\mu, \lambda)$ which denotes the infimum of the Lagrangian. Then we can build the dual problem:

$$\begin{aligned} \max \quad & \psi(\mu, \lambda) \\ \text{s.t.} \quad & \lambda \geq 0 \end{aligned} \tag{11}$$

It aims to obtain the greatest lower bound on the solution to the primal problem. With the dual problem, solving the optimisation problem becomes easier.

We then apply dual ascent techniques to approach the dual problem, updating the Lagrangian multiplier λ by maximising $\psi(\mu, \lambda)$ and updating μ by minimising $L(\mu, \lambda)$ interchangeably by their gradients, as the following:

$$\lambda = \lambda + \alpha \nabla_\lambda L(\mu, \lambda),$$

$$\mu = \arg\min_\mu L(\mu, \lambda),$$

where α is the learning rate.

Algorithm 4 Dual Ascent

Input: Training set $\vec{b}'$, budget β, learning rate α, number of episodes T
Output: Allocation function q^μ
1: Set $t = 0$, and initialise λ, q^μ. Set $BF = False$
2: **while** $t \leq T$ **do**
3: Generate allocation vector $(q_1, \cdots, q_n) = q^\mu(\vec{b}')$
4: Compute Lagrangian $L(\lambda, \mu)$
5: Let $\mu' = \arg\min_\mu L(\lambda, \mu)$
6: **if** BF constraint (7) is satisfied **then**
7: Set $\mu = \mu'$, $BF = True$
8: **end if**
9: $\lambda = \lambda + \alpha \nabla_\lambda L(\mu, \lambda)$, $t = t + 1$
10: **end while**
11: Return q^μ if BF, NO ANSWER otherwise

To be specific, the dual ascent, as shown in Algorithm 4, takes the training set, budget, learning rate and the number of episodes as input, where the training set is generated from the same distribution and has the same size as the real bids of data owners. The algorithm initialises t, λ and q^μ (Line 1 of Alg. 4). Then for each episode, the algorithm passes the training set to the allocation function q^μ and generates the allocation vectors used to compute the Lagrangian (Line 3 of Alg. 4). The algorithm updates the parameters of q^μ using the SGD method to find μ' that minimises the Lagrangian, which can be treated as the loss function in training q^μ (Line 4 of Alg. 4). Then it updates μ only if the allocation vector generated by $q^{\mu'}$ and the corresponding payment vector satisfy the BF constraint (Line 5 of Alg. 4), which ensures that the trained allocation function and the corresponding payment vector always satisfy the BF constraint. The algorithm updates λ using the gradient of the (updated) Lagrangian (Line 6 of Alg. 4). Then it increments t (Line 7 of Alg. 4) and continues the training until finishing all the episodes. Finally, the algorithm outputs a trained allocation function q^μ.

After training the neural-based allocation function q^μ, NPQM follows the proposed framework step by step (Line 1-7 of Alg. 1), generating the allocation vector $(q_1, \cdots, q_n) = q^\mu(\vec{b})$ and computing the expected payments $(P_1, \cdots, P_n)$, where $P_i = b_i w_i(b_i) + \int_{b_i}^{\overline{\theta}} w_i(x)dx$, $w_i(b_i) = q_i(b_i) \ln \frac{1+q_i(b_i)}{1-q_i(b_i)}$. The ILR $\mathcal{L}_I$, illustrated in Alg. 2, is applied to each data owner s_i, computing the perturbed private data t_i' and the compensation p_i. Finally, the query is answered based on the collected dataset $\{t_1', \cdots, t_n'\}$ and the payment vector and query answer are returned. The details are illustrated in Alg. 5.

Lemma 5 *NPQM that employs q^μ as the allocation function is IC and IR.*

Theorem 6 *The neural-based private query mechanism (NPQM) is IC, IR, BF and ε_i-LDP.*

Algorithm 5 Neural-based Private Query Mechanism (NPQM)

Input: Data owners $\vec{s}$, budget β, query f, allocation function q, training set $\vec{b}'$, learning rate α, number of episodes T
Output: Payment vector $(p_1, \ldots, p_n)$, query answer z
1: Train the allocation function $q = DualAscent(\vec{b}', \beta, \alpha, T)$
2: Generate the allocation vector $(q_1, \ldots, q_n) = q(\vec{b})$
3: Generate the payment vector $(P_1, \ldots, P_n)$, where $P_i = b_i w_i(b_i) + \int_{b_i}^{\overline{\theta}} w_i(x)dx$
4: **for** each data owner s_i **do**
5: Get a random value and the payment $t_i', p_i = \mathcal{L}_I(t_i, q_i, P_i)$
6: **end for**
7: Compute the query answer $z = f(t_1', \ldots, t_n')$
8: Return $(p_1, \ldots, p_n), z$

Please see the full paper for the proof of Lemma 5 and Theorem 6.

7 Experiments

7.1 Experiment setup

In experiments, we aim to evaluate the accuracy of GPQM and NPQM when applied to various query types and datasets. The experiment setup is summarised in Table 1.

Table 1: Experiment Setups

Query f	Count, median
Dataset D	Obesity, Maternal, Exam, Students, Salaries, Customers
Budget β	$\{0.1n, \cdots, 0.9n\}$
Bids $\vec{b}$	Drawn from $U(0, 1)$
Mechanism	GPQM (linear, log, exp), NPQM, FQ mechanisms

Datasets: We use six real-world datasets, Obesity [31], Maternal Health Risk (Maternal) [2], Exam [21], Students' Dropout and Academic Success (Students) [27], Data Science Salaries 2023 (Salaries) [5] and Customer Personality Analysis (Customers) [3], as the private data of data owners. See more details in the full paper.

Bids $\vec{b}$: The bids of data owners are generated from the uniform distribution $U(0, 1)$. In words, the range of bids $(\underline{\theta}, \overline{\theta}) = (0, 1)$.

Budget: The budget of consumer is set to be $0.1\overline{\theta}n$ to $0.9\overline{\theta}n$.

Queries: We analyse count and median queries. For count queries, we utilise the obesity level from Obesity, the risk level from Maternal, the test preparation status from Exam, the scholarship holder from Students, the remote working ratio from Salaries, and the complaint of Customers. Specifically, we count the number of overweight individuals, high-risk pregnant females, students who prepared for the exams, students who hold a scholarship, people who work remotely, and customers who made complaints.

For median queries, we use the age of Obesity and Maternal, and the reading test score of Exam, the admission test score of Students, the salary in US dollars of Salaries, and in-store shopping of Customers. Specifically, we query the median age of individuals in Obesity and Maternal, the median score in the reading test, the median salary in US dollars, and the median number of in-store shopping.

Mechanims: We evaluate five mechanisms: three implementations of GPQM, NPQM, and FairQuery (FQ) [20]. For GPQM, we consider linear, logarithmic, and exponential allocation functions, denoted by $q(\vec{b}) = 1 - \vec{b}$, $q(\vec{b}) = -\log(k_l q(\vec{b}))$, and $q(\vec{b}) = e^{-k_e q(\vec{b})}$, where the coefficients k_l, k_e are randomly selected from the uniform distribution $U(0, 10)$ for each query. The implementation details of NPQM are shown in Appendix D of our full paper.

We make modifications to the FQ mechanism [20] to ensure LDP for count and median queries, and we consider this modified version

as our benchmark (see Appendix A of our full paper).In the modified FQ, the procurement process remains unchanged such that the data owners are sorted in descending order based on the value of b_i, and the mechanism determines the last data owner k who can be selected within the given budget. Additionally, we introduce modifications to the query process by incorporating a widely used local randomiser *randomised response* [35] that allows data owners to report their true data with a probability of q_i and report a random value otherwise.

Experiments: For each experiment, we generate bids of data owners $\vec{b}$ randomly from the uniform distribution $U(0, 1)$. We conduct $m = 100$ queries for each mechanism and budget. For count queries, the performance of mechanisms is measured under average relative absolute error (RAE), i.e., $\frac{1}{m}\sum_{i=1}^{m}\frac{|z-z_g|}{z_g}$, where z denotes the query answer generated by mechanisms and z_g denotes the ground truth. For median queries, the performance is measured under average absolute error (AE), i.e., $\frac{1}{m}|z - z_g|$.

Implementation details: NPQM is implemented in Python 3.9 on NVIDIA GeForce RTX 3090 Ti GPU. All mechanisms are implemented in Python 3.9 on Apple M1 Pro CPU. The code is available on https://anonymous.4open.science/r/IntegratedPDQS

7.2 *Results and discussion*

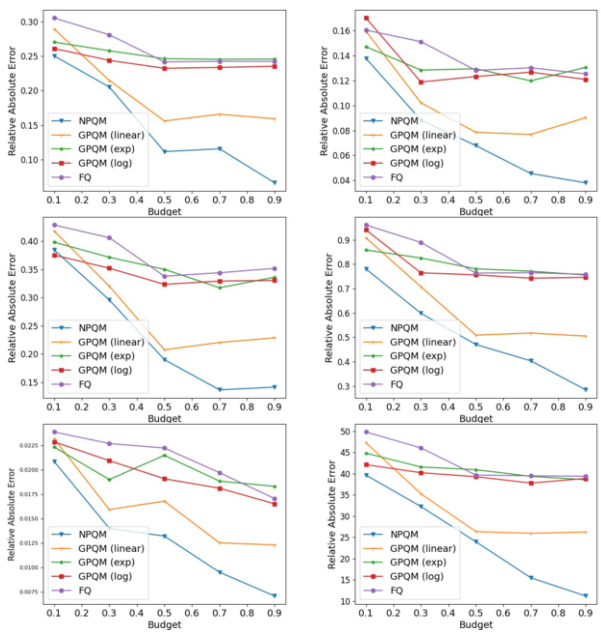

Figure 2: Relative absolute error of count queries on the Obesity (top-left), Maternal (top-right), Exam (middle-left), Students (middle-right), Salaries (bottom-left) and Customers (bottom-right) datasets

We evaluate and compare the performance of the five mechanisms. Fig 2-3 display the RAE and confidence intervals of the mechanisms for count queries on the six datasets, respectively. The AE and confidence intervals for median queries are detailed in the full paper.

In most experiments, our approach outperforms the benchmark. NPQM achieves the best accuracy on both count and median queries, with the lowest RAE / AE and the estimated query answer closest to the ground truth. GPQM with a linear allocation function also performs well. GPQM with exponential and logarithmic allocation functions outperforms the benchmark in most experiments when the budget is below $0.5n$, but this advantage becomes less noticeable when the budget exceeds $0.5n$.

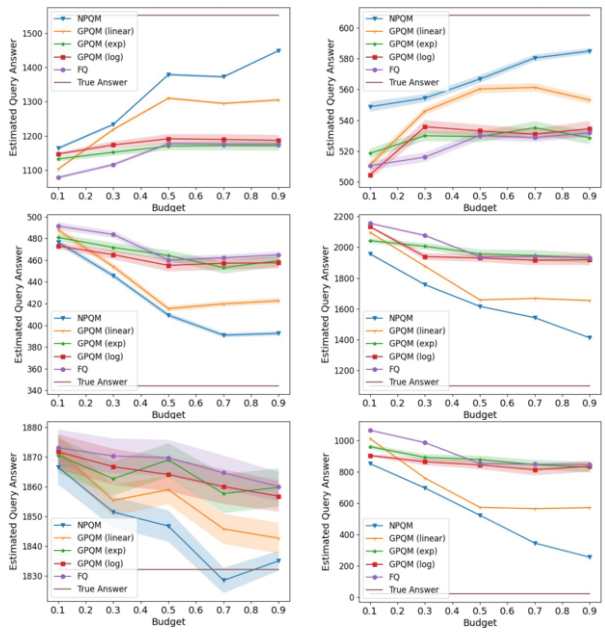

Figure 3: Confidence interval of the estimated answer of count queries on the Obesity (top-left), Maternal (top-right), Exam (middle-left), Students (middle-right), Salaries (bottom-left) and Customers (bottom-right) datasets

As expected, the performance of NPQM and GPQM with linear allocation improves as the budget increases. The advantages of NPQM and GPQM with linear allocation become more significant compared to other mechanisms as the budget increases, indicating that these mechanisms can approximate the ground truth with a sufficient budget. On the other hand, for GPQM with exponential and logarithmic allocation and FQ, their accuracy does not improve further as the budget reaches $0.5n$.

The experimental results demonstrate that NPQM consistently performs better and provides accurate estimates for various query types and datasets. NPQM also performs well when bids follow a normal distribution, and even when the training and test sets come from different distributions (see Appendix E of our full paper). Despite the potential budget overruns from using expected payment as a training constraint, NPQM remains an accurate solution, as such occurrences are rare and can be resolved through system re-runs. Besides, we observe that GPQM's performance relies on the choice of the allocation function. As for our benchmark, it suffers from limitations as it needs to allocate a portion of the budget to data owners who cannot truthfully report their private data, resulting in compromised accuracy even with a sufficient budget.

8 Conclusion

We introduce an integrated PDQS framework, which combines the procurement and query processes, and effectively utilises the consumer's budget to approximate query accuracy under LDP. We propose two implementations of the novel framework, GPQM and NPQM, which address queries while considering IC, IR, and BF constraints. The experimental results demonstrate that our mechanisms outperform existing approaches that separate the procurement and query processes in query accuracy. Potential future work can be extending the Integrated PDQS framework to handle sequential data and multi-dimensional private data with varying privacy requirements for different dimensions.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China No. 62172077.

References

[1] Jacob D Abernethy, Rachel Cummings, Bhuvesh Kumar, Sam Taggart, and Jamie H Morgenstern, 'Learning auctions with robust incentive guarantees', *Advances in Neural Information Processing Systems*, **32**, (2019).

[2] Marzia Ahmed and Mohammod Abul Kashem, 'Iot based risk level prediction model for maternal health care in the context of bangladesh', in *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1–6. IEEE, (2020).

[3] Akash Patel. Customer personality analysis. https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis, Accessed 2023.

[4] Aaron Archer and Éva Tardos, 'Truthful mechanisms for one-parameter agents', in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 482–491. IEEE, (2001).

[5] Arnab Chakraborty. Data science salaries 2023. https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023, Accessed 2023.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al., 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Foundations and Trends® in Machine learning*, **3**(1), 1–122, (2011).

[7] Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou, 'Optimum statistical estimation with strategic data sources', in *Conference on Learning Theory*, pp. 280–296. PMLR, (2015).

[8] Xi Chen, Sentao Miao, and Yining Wang, 'Differential privacy in personalized pricing with nonparametric demand models', *Operations Research*, (2022).

[9] Yiling Chen, Nicole Immorlica, Brendan Lucier, Vasilis Syrgkanis, and Juba Ziani, 'Optimal data acquisition for statistical estimation', in *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 27–44, (2018).

[10] Yiling Chen and Shuran Zheng, 'Prior-free data acquisition for accurate statistical estimation', in *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 659–677, (2019).

[11] Rachel Cummings, Hadi Elzayn, Vasilis Gkatzelis, Emmanouil Pountourakis, and Juba Ziani, 'Optimal data acquisition with privacy-aware agents', *arXiv preprint arXiv:2209.06340*, (2022).

[12] Rachel Cummings, Katrina Ligett, Aaron Roth, Zhiwei Steven Wu, and Juba Ziani, 'Accuracy for sale: Aggregating data with a variance constraint', in *ITCS 2015*, pp. 317–324, (2015).

[13] Pranav Dandekar, Nadia Fawaz, and Stratis Ioannidis, 'Privacy auctions for recommender systems', *ACM Transactions on Economics and Computation (TEAC)*, **2**(3), 1–22, (2014).

[14] John C Duchi, Michael I Jordan, and Martin J Wainwright, 'Local privacy and statistical minimax rates', in *FOCS 2013*, pp. 429–438. IEEE, (2013).

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, 'Calibrating noise to sensitivity in private data analysis', in *Theory of cryptography conference*, pp. 265–284. Springer, (2006).

[16] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova, 'Rappor: Randomized aggregatable privacy-preserving ordinal response', in *CCS 2014*, pp. 1054–1067, (2014).

[17] Alireza Fallah, Ali Makhdoumi, Azarakhsh Malekian, and Asuman Ozdaglar, 'Optimal and differentially private data acquisition: Central and local mechanisms', in *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 1141–1141, (2022).

[18] Lisa K Fleischer and Yu-Han Lyu, 'Approximately optimal auctions for selling privacy when costs are correlated with data', in *EC 2012*, pp. 568–585, (2012).

[19] Arpita Ghosh, Katrina Ligett, Aaron Roth, and Grant Schoenebeck, 'Buying private data without verification', in *EC 2014*, pp. 931–948, (2014).

[20] Arpita Ghosh and Aaron Roth, 'Selling privacy at auction', in *EC 2011*, pp. 199–208, (2011).

[21] Jakki Seshapanpu. Students performance in exams. https://www.kaggle.com/datasets/spscientist/students-performance-in-exams, Accessed 2023.

[22] Fengjuan Jia, Mengxiao Zhang, Jiamou Liu, and Bakh Khoussainov, 'Incentivising diffusion while preserving differential privacy', in *Uncertainty in Artificial Intelligence*, pp. 963–972. PMLR, (2023).

[23] Yanzhe Murray Lei, Sentao Miao, and Ruslan Momot, 'Privacy-preserving personalized revenue management', *Sentao and Momot, Ruslan, Privacy-Preserving Personalized Revenue Management (October 3, 2020)*, (2020).

[24] Katrina Ligett and Aaron Roth, 'Take it or leave it: Running a survey when privacy comes at a cost', in *WINE 2012*, pp. 378–391. Springer, (2012).

[25] Yang Liu and Yiling Chen, 'Learning to incentivize: Eliciting effort via output agreement', *arXiv preprint arXiv:1604.04928*, (2016).

[26] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, et al., *Big data: The next frontier for innovation, competition, and productivity*, McKinsey Global Institute, 2011.

[27] Mónica V Martins, Daniel Tolledo, Jorge Machado, Luís MT Baptista, and Valentim Realinho, 'Early prediction of student's performance in higher education: A case study', in *Trends and Applications in Information Systems and Technologies: Volume 1 9*, pp. 166–175. Springer, (2021).

[28] Frank McSherry and Kunal Talwar, 'Mechanism design via differential privacy', in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, (2007).

[29] Kobbi Nissim, Claudio Orlandi, and Rann Smorodinsky, 'Privacy-aware mechanism design', in *Proceedings of the 13th ACM conference on electronic commerce*, pp. 774–789, (2012).

[30] Kobbi Nissim, Salil Vadhan, and David Xiao, 'Redrawing the boundaries on purchasing data from privacy-sensitive individuals', in *Proceedings of the 5th conference on Innovations in theoretical computer science*, pp. 411–422, (2014).

[31] Fabio Mendoza Palechor and Alexis de la Hoz Manotas, 'Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico', *Data in brief*, **25**, 104344, (2019).

[32] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson, 'Monotonic value function factorisation for deep multi-agent reinforcement learning', *The Journal of Machine Learning Research*, **21**(1), 7234–7284, (2020).

[33] Aaron Roth and Grant Schoenebeck, 'Conducting truthful surveys, cheaply', in *EC 2012*, pp. 826–843, (2012).

[34] Weina Wang, Lei Ying, and Junshan Zhang, 'Buying data from privacy-aware individuals: the effect of negative payments', in *WINE 2016*, pp. 87–101. Springer, (2016).

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

[36] Haifei Yu and Mengxiao Zhang, 'Data pricing strategy based on data quality', *Computers & Industrial Engineering*, **112**, 1–10, (2017).

[37] Mengxiao Zhang and Fernando Beltrán, 'An experimental study on discovering the value of private data in data marketplaces', *Available at SSRN 4053024*, (2022).

[38] Mengxiao Zhang, Fernando Beltran, and Jiamou Liu, 'Selling data at an auction under privacy constraints', in *UAI 2020*, pp. 669–678, (2020).

[39] Mengxiao Zhang, Fernando Beltrán, and Jiamou Liu, 'A survey of data pricing for data marketplaces', *IEEE Transactions on Big Data*, (2023).

[40] Mengxiao Zhang, Jiamou Liu, Kaiyu Feng, Fernando Beltran, and Zijian Zhang, 'Smartauction: A blockchain-based secure implementation of private data queries', *Future Generation Computer Systems*, **138**, 198–211, (2023).

[41] Mengyuan Zhang, Lei Yang, Xiaowen Gong, and Junshan Zhang, 'Privacy-preserving crowdsensing: Privacy valuation, network effect, and profit maximization', in *GLOBECOM 2016*, pp. 1–6. IEEE, (2016).