Protecting Data Markets from Strategic Buyers

Raul Castro Fernandez
raulcf@uchicago.edu
The University of Chicago

ABSTRACT

The growing adoption of data analytics platforms and machine learning-based solutions for decision-makers creates a significant demand for datasets, which explains the appearance of data markets. In a well-functioning data market, sellers share data in exchange for money, and buyers pay for datasets that help them solve problems. The market raises sufficient money to compensate sellers and incentivize them to keep sharing datasets. This low-friction matching of sellers and buyers distributes the value of data among participants. But designing online data markets is challenging because they must account for the strategic behavior of participants.

In this paper, we introduce techniques to protect data markets from strategic participants, even when the asset traded is data. We combine those techniques into a pricing algorithm specifically designed to trade data. The evaluation includes a user study and extensive simulations. Together, the evaluation demonstrates how participants strategize and the effectiveness of our techniques.

CCS CONCEPTS

• **Information systems** → *Data management systems*; *Collaborative and social computing systems and tools.*

KEYWORDS

data markets; value of data; strategic participants

ACM Reference Format:
Raul Castro Fernandez. 2022. Protecting Data Markets from Strategic Buyers. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/XXXXXXX.YYYYYY

1 INTRODUCTION

In an increasingly data-driven world, access to data is more valuable than ever before. The growing adoption of data analytics platforms and machine learning-based solutions for decision-makers creates a significant demand for datasets, and this has led to the appearance of data markets [16, 27]. Current data markets [8, 21, 62, 63] work as a storefront showing a list of datasets for sale, but data transactions involve tedious one-off negotiations between sellers and buyers. Worse, datasets' prices are set in an ad-hoc manner and as a result, sellers risk losing revenue, and buyers risk overpaying. In a well-functioning data market, sellers share data in exchange for money, and buyers pay for datasets that help them solve problems. The market raises sufficient money from buyers to compensate sellers for their data and incentivize them to share more datasets. A key challenge to designing these low-friction matching of sellers and buyers is to *price* data in a way that maximizes data transactions that distribute the value of data.

In well-functioning markets of common goods and services, prices are set based on the forces of supply and demand. Unfortunately, the methods used to price common goods and services do not apply directly to data [4] due to its unique characteristics:

• **Data's combinatorial power.** Market mechanisms for common goods rely on knowledge about the consumers' willingness to pay for the good to set a price. Such knowledge is obtained via market research techniques [58]. Unlike common goods, data can be used for many different purposes, each with different value for the beneficiary. And because data can be combined with other datasets, its uses are numerous and hard to quantify. Together, this means that it is not possible to assume a priori knowledge of consumer's willingness to pay for data. As an asset, data resembles more a piece of art, an expensive bottle of wine, or diverse antiques (that are hard to price due to lack of knowledge of consumers' willingness to pay) than it resembles cars, houses or other digital goods such as a movie rental or a concert ticket, where market research will offer insights on consumers' willingness to pay.

• **Data is nonrival and easy to replicate.** Unlike antiques, art, or bottles of wine which can be allocated to a consumer once (i.e., they are rival goods), data can be used by all consumers at once, i.e., it is a nonrival asset more similar to an idea. Furthermore, because data is easy to replicate, it is possible to make data available to everyone who benefits from it at low marginal cost. What this means is that the auction techniques [50] used successfully to trade rival assets with unknown consumer's willingness to pay such as antiques and art cannot be directly used with data. Although there are techniques to sell digital goods (with infinite supply) these are meant to sell those goods once. A single dataset can be sold multiple times over time (e.g., as part of different data combinations) for different purposes with different valuations.

In this paper, we design pricing algorithms for data markets. Although the pricing algorithm could in principle be used to trade any asset, that is undesirable because there are better pricing methods to trade common goods and services. The opposite is not true, pricing algorithms for common assets do not apply to data due to the characteristics described above.

In designing a pricing algorithm for data, we want to set prices for seller-provided datasets based on the demand generated by interested buyers via submitted *bids*. Because we do not know buyers willingness to pay, we cannot use traditional auction theory that relies on that knowledge. Instead, we resort to prior-free techniques,

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

ACM ISBN XXXX.YYYY.ZZZZ…$15.00
https://doi.org/10.1145/XXXXXXX.YYYYYY

such as those that learn prices online, as bids are received [35]. In this paper, we focus on the design of protection techniques to shield the pricing algorithm from *bid*s submitted by strategic buyers that try to improve their benefits. In doing so, the paper complements the growing literature on data markets in data management by drawing attention to the need for protection from strategic behavior. To address strategic behavior, we propose a pricing algorithm for buyer-seller markets of the kind described above that tackles the following challenges:

• Because price is set based on bids generated by human buyers, these may send artificially low bids to overfit the price in a profitable direction. We introduce Epoch-Shield to protect against these artificially low bids.

• Buyers benefit from a dataset as long as they obtain it within a time period, e.g., if they buy the data earlier than the time it takes them to collect it or prepare it manually. Some buyers will strategize over time within the full period, e.g., bidding low to drive the prices of datasets they want down just before the submit a real bid. We introduce Time-Shield to protect against this behavior.

• Buyers will not always choose the optimal action i.e., they make boundedly-rational choices. This is important to account for because non-optimal decisions can hurt the market. We introduce Uncertainty-Shield to protect against boundedly-rational behavior.

We combine the above techniques into a pricing algorithm. We conduct an IRB-approved user study to: i) motivated the need to protect data markets; ii) to demonstrate the effectiveness of the protection techniques. We complement the user study with extensive simulations that study and show the effectiveness of our techniques in different scenarios. We discuss an alternative implementation of the pricing algorithm that uses differential privacy (Section 6.3). Finally, we discuss an ex-post version of the algorithm (Section 8) that works even when buyers do not know the dataset valuation before using the data, i.e., when data is an experience good.

We organize the rest of the paper as follows. Section 2 presents the market model. Sections 3, 4, and 5 introduce the protection techniques, followed by their integration in the pricing algorithm (Section 6). We then present the evaluation results in Section 7 followed by a discussion of the ex-post algorithm in Section 8, and related work in Section 9. We present the conclusions in Section 10.

2 DATA MARKET MODEL

In this section, we present the market model (Section 2.1), followed by the data buyer model in Section 2.2, and the arbiter model in Section 2.3, before explaining the challenges of designing data markets when buyers are strategic in Section 2.4.

2.1 Data Market Preliminaries

A market consists of buyers who want to buy datasets, sellers who want to share datasets in exchange for a reward such as money, and an arbiter, as shown in Fig. 1.

The arbiter's goal is to generate transactions that satisfy the buyers' requests with the seller's supplied datasets. It needs to achieve this while maximizing the revenue extracted from buyers to compensate sellers and incentivize them to keep sharing data. Achieving this goal requires generating transactions. To do so, the arbiter may

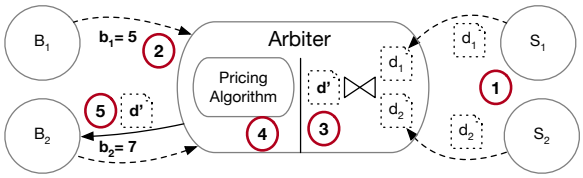

Figure 1: Market Model. B are buyers and S are sellers. They exchange datasets via an Arbiter that finds data appropriate for buyers, and prices it according to its demand.

combine datasets uploaded by sellers to expand the offering beyond individual datasets and hence the likeliness of satisfying the buyers' needs (step 3 in Fig. 1)[1]. Buyers declare their interest for a desired dataset d' (see the figure) with a bid, as illustrated in step 2 of the figure (we explain below how they generate the bid). This bid propagates[2] to the datasets used to produce d', which in the example correspond to d_1 and d_2. As a result of the previous flow, the same dataset may participate in multiple different combined datasets to satisfy the needs of the same or different buyers over time. This means that the market cannot restrict buyers to bid only once because that would significantly limit the market's potential utility [37]. The consequence of multiple buyers participating in the market is the generation of a stream of bids that apply to the datasets uploaded by sellers. In addition to generating transactions, the arbiter's goal is to find a way of pricing the datasets based on the bids so as to maximize revenue, incentivizing sellers to keep sharing datasets.

Sellers share data, d, with the arbiter in exchange for money (1 in the Figure). We do not include data licensing or sharing agreements in our model because they do not affect the protection mechanisms we introduce. We assume dataset exchange does not create externalities [3], i.e., no personal or private information is traded. This assumption is consistent with the major data markets today [8, 62, 63] which do not permit trading personal information.

Buyers have varied data management needs, from augmenting a training dataset with additional features and samples, to obtaining a curated dataset useful to complete a report, and many others. Similarly, buyers may require datasets with specific formats, with specific quality (e.g., no missing values) and other intrinsic properties they find valuable. These requirements become part of the specific data management need. We abstract away the specific use cases by assuming buyers can link the business value of solving their data management tasks to a valuation, v_i, which is private information only they know[3]. Then, buyers send a *bid*, b_i to the market indicating how much they are willing to pay to obtain d' (2 in Fig. 1). This permits the design of a conceptually simple and general market model. Because d' can be sold infinite times buyers may try to send false bids, $b_i \leq v_i$, also called *non-truthful* or *strategic* bids. Buyers would send these strategic bids hoping they can still obtain d' at a lower price. Buyers' strategic behavior imposes another requirement on the arbiter, who must protect the pricing mechanism against strategic behavior.

[1] We do not deal with the problem of combining datasets, we assume manual labor or integration solution is available

[2] We do not deal with the problem of bid propagation in this paper but note that it can be solved using provenance techniques.

[3] We break the assumption of buyers knowing the valuation a priori in Section 8

Scope of this paper. There are many types of market attacks, including *strategic*, *exploitation*, and *collusion* attacks. The attack may originate from buyers, arbiter, and sellers. In this paper we focus on buyers only. *Exploitation* attacks include denial of service attacks and false-name bidding, where buyers create multiple identities. Deterrent mechanisms against exploitation attacks include legal and technical mechanisms. For example, a technical mechanism to prevent false-name bidding is to bind bids to buyers via a signature scheme that requires a proof of identity. In *collusion* attacks, multiple different buyers form coalitions outside the market. Note that a single buyer may use false-name bidding to form a coalition, but we addressed false-name bidding above. Coalitions can be harmful when they share the information they learn from interacting with the market, when they collaborate to influence prices in a profitable direction, and when they cost-share buying data and then exchange it internally. Mechanisms to deal with exploitation attacks are orthogonal to the techniques we introduce in this paper, so we assume a market deployment implements those techniques and hence there are no false-name bids in our setup. A full treatment of collusion is out of the scope of this paper. As many others, we assume collusion does not take place [32, 33]. The scope of this paper is the design of a pricing algorithm for data markets that is robust to buyers' strategic behavior.

2.2 Buyer Model: Patience and Valuation

A buyer i wants to obtain data, d, to increase its utility, u_i. Buyer's utility depends on: i) whether they get d as determined by the *allocation function* X; ii) the difference between the price paid for d, $p_t(d)$ and the buyer's private valuation v_i; and iii) *when* the dataset is allocated. u_i is defined as follows:

$$u_i(v_i, b_i, t, d, \tau_i) = \delta(\tau_i, t) * X(b_i, p_t(d)) * (v_i - p_t(d)) \quad (1)$$

$t \in [0, ...)$ indicates time and increases monotonically. The *allocation decision*, $X_i = 1$[4] when $b_i > p_t(d)$, i.e., the bid is higher than d's price at time t. $X_i = 0$ when buyers lose the allocation, in which case $u_i = 0$. The *deadline-patience function*, $\delta(\tau_i, t)$, takes value 1 when $t \leq \tau_i$ and 0 otherwise. This function represents scenarios where a dataset is only useful to the buyer if obtained before a deadline, $\tau_i \in [0, ...)$. The buyer's deadline, τ_i is private information to the buyer and unknown to the arbiter. When $X_i = 1$, and the allocation takes place before the deadline, τ_i, the utility is the difference between the buyer's private valuation, v_i (which is how much value the buyer associates with d), and the price of d at time t, which is determined by $p_t(d)$. The larger the difference between v_i and $p_t(d)$, the larger the utility. The market implementation must ensure there are no pending bids after a buyers' deadline has passed. Our proposal and algorithm assumes the market implementation correctly achieves this goal.

Deadline-patience utility function: Many real data management tasks require access to integrated datasets. An integrated dataset is a combination of other existing datasets, prepared for a specific task such as training a machine learning model or produce a report from a business intelligence tool. Integrating datasets is a time-consuming and tedious process [28]. With this context, consider a data analyst facing the decision of investing time on integrating datasets manually to produce the desired dataset, or sourcing an appropriate dataset externally [27], for example, by acquiring it in a data market. For the analyst to benefit from the market, it must obtain the dataset in less time than what it would take to prepare it manually, and that provides a deadline, τ_i, within which the market yields utility to the analyst, i.e., who plays the role of a buyer when interacting with the data market. We note that the approach presented in the paper supports other patience functions, such as those that would progressively decrease the utility for the buyer; we stick to the simpler to define yet realistic deadline-function because it simplifies the subsequent analysis.

Interpreting Buyer's Private Valuation. A buyer's utility depends on its private valuation, v_i, which the buyer may know before bidding or not. We consider both scenarios below.

Buyers know their valuation in advance. Sometimes buyers can link dataset to business value, so they know how to value such a dataset, i.e., they know their private valuation v_i. In this case, buyers communicate their value to the arbiter via a bid, b_i.

Buyers know v_i after using the data, *ex-post*. Sometimes, data is an experience good [19] because its value to the buyer is not known until after data has been used for a task. For example, this happens when a buyer wants to engage in an exploratory task using a dataset but does not know how much value will be extracted from such activity a priori, and hence they do not know v_i. We assume, however, that buyers learn their v_i after using the data, and we discuss a technique in Section 8 that supports data trading under this challenging scenario. Until then, we assume the common scenario where buyers know their valuation.

2.3 Arbiter Model and Posting Price

The arbiter must allocate data to buyers to maximize revenue that is used to compensate and incentivize sellers to keep sharing data. The arbiter could allocate data to anybody who asks because data is non-rival, but then buyers would bid low knowing that they will receive the data anyway, and consequently, the arbiter would not raise sufficient revenue to motivate sellers to share.

Overview of mechanism design. Mechanism design [52] studies the design of allocation and pricing functions to optimize a market objective such as welfare or revenue. When the buyer's value distribution is known a priori, such as often assumed in the Economics literature [50], the design of auctions and posting price mechanisms is well understood even for revenue maximization scenarios. For example, the performance of a second price auction with reserve price is well characterized[35] when buyer's value distribution is well known and can be used to calculate the reserve price. However, when buyer's value distribution are not known (such as when selling data) the options narrow down to prior-independent and prior-free approaches. Among those, we highlight techniques based on online learning (and regret minimization) that try to learn the value distribution on the fly [13, 40]. The methods we use in this paper have this online learning flavor, but our focus is on protecting them from strategic buyers, as we explain next.

Posting Prices. One approach to make allocation decisions that takes into consideration the demand for data is to choose a *posting*

[4] X is the allocation function and X_i is the allocation decision for buyer i

price, p, given by the pricing function $p_t(d)$, before receiving bids for d. Bids higher than the posting price win and winners pay the posting price [42]. With this mechanism, buyers face a simple action space, where bidding their true valuation is both easy and their best bet to win the allocation. Still, some buyers will try to strategize. Using this approach the arbiter makes allocation decisions as bids arrive, hence returning an answer to buyers with low latency, and hence, not affecting negatively their utility, which depends on *when* they obtain the dataset due to the patience parameter.

The challenge of posting price mechanisms is choosing p before the bids arrive. We call *update algorithm*, $a(\vec{b})$, to the algorithm that chooses $p_t(d)$ based on a vector of bids, $\vec{b}$. Before discussing design options for $a(\vec{b})$, it is helpful to consider first the optimal posting price if all bids were known in advance.

Optimal Posting Price: Consider an arbiter wants to choose p to maximize revenue in *one round* of bids; this is a *digital goods auction* [11, 13, 34]. Suppose the buyer's bids, $\vec{b}$ are known in advance, and let b_k be the *k-th* largest bid in $\vec{b}$. Then, an algorithm maximizes revenue when it chooses k such that $k * b_k$ is maximum:

$$M(\vec{b}) = \max_k(k * b_k) \tag{2}$$

According to this algorithm, $p_t(d) = b_k$ (break ties by choosing larger b_k). In this case, all buyers with $b_i \geq b_k$ win and obtain the data by paying b_k. The rest of the buyers lose and do not obtain the data. The arbiter raises $k * b_k$ in revenue. This ideal mechanism works when bids are known in advance.

Online Posting Price: In data markets, bids, $\vec{b}$, arrive in a streaming fashion. It is possible to use online algorithms [11, 13, 34, 40] that use past bids to implement $a(\vec{b})$ and calculate p with good revenue guarantees. Unfortunately, online mechanisms will be gamed by buyers, so they must be protected from buyers' strategies.

2.4 Challenges of Pricing in Data Markets

Buyers know the *update algorithm* used by the arbiter to set p because we do not want to protect the market through obscurity, akin to the rejection of the *security by obscurity* principle in security engineering [53]. Then, because buyers know $p_t(d)$ is chosen based on past bids, buyers can send artificially low bids to overfit the algorithm in a direction that is profitable for them. To protect against this behavior we introduce Epoch-Shield (Section 3). Buyers will react to the existence of Epoch-Shield and strategize by exploiting all the time they have available before their deadline expires to obtain the data. To protect against these strategic bids over time, we introduce Time-Shield (Section 4). Finally, some buyers may bid erratically, harming market's revenue: we protect the market from this behavior with Uncertainty-Shield (Section 5).

3 EPOCH-SHIELD PROTECTION

In this section, we present Epoch-Shield, a technique to protect data markets from strategic low bids. Buyers will react to Epoch-Shield, however, and that is why we need the Time-Shield protection mechanism of the next section to complement Epoch-Shield.

3.1 Preliminaries

Two-Round Posting Price Mechanism is not Truthful. Consider a buyer who knows the arbiter uses $a(\vec{b})$ to choose $p_t(d)$ and has two opportunities to bid, at time t_1 and t_2. The prices chosen by $a(\vec{b})$ at times t_1 and t_2 are p_1 and p_2, respectively. To maximize its utility, the buyer bids non-truthfully $b_i \leq v_i$ at t_1, hoping that $a(\vec{b})$ will adapt p_2 in order to reflect the new demand, making $p_2 \leq p_1$. At t_2, the buyer bids truthfully, $b_i = v_i$. If its first bid successfully drove the price down, then buyer's utility increases because $v_i - p_2 \geq v_i - p_1$. This strategic behavior harms the market's revenue because the arbiter does not know if buyers are truthful.

3.2 The Epoch-Shield Technique

The insight buyers exploit to game the market is that $a(\vec{b})$ will adapt $p_t(d)$ based on the incoming bids. Our goal is to design $a(\vec{b})$ so that it is hard to game. This is achieved by designing $a(\vec{b})$ so no single bid influences the price by much.

Epoch-Shield: Computing on epochs instead of individual bids. Instead of updating $p_t(d)$ using the latest bid, it is updated using the last $E \geq 1$ bids. We call E epoch. An epoch *logically* splits the incoming sequence of bids into disjoint groups of E bids, akin to how a window splits an event stream in stream processing [61].

The posting price is updated once per epoch, and new incoming bids are evaluated against the latest posting price. Buyers do not know the epoch size and therefore do not know when posting prices are updated. By choosing $p_t(d)$ based on E, the arbiter can choose a function to smooth out the effect of any single bid based on an *update algorithm*, $a(\vec{b})$.

Update algorithm. When choosing an update algorithm, we wish two properties: i) that a single bid's influence on the function's output is not significant; ii) that it is not clear, from the buyer's perspective, what bid drives the posting price in the desired direction. For example, a simple average, $a(\vec{b}) = avg(\vec{b})$ is not appropriate because it is sensitive to outliers, i.e., low bids. A robust statistic, such as the median, is preferable because the influence of a single bid is lower. However, from the perspective of a strategic buyer, it is clear that bidding low is strictly helping its cause by moving the median in the desired direction.

To make it hard for buyers to influence the price, the arbiter computes $p_t(d)$ for the next epoch based on the bids collected in the current epoch according to Equation 2, so $p_t(d) = b_k$. The advantage of using this algorithm is that buyers are no longer guaranteed to influence $p_t(d)$ by bidding low because finding the bid that optimizes the influence on $p_t(d)$ is hard. In particular, to game the market, a buyer must find a b_i^* so that:

$$\arg\min_{b_i^*} \; r(\vec{b}||b_i^*) \;\text{ s.t. }\; r(\vec{b}||b_i^*) > r(\vec{b}),\;\; pr(\vec{b}||b_i^*) < pr(\vec{b}) \tag{3}$$

where $r(\vec{b})$ is a function that returns the optimal revenue, $pr()$ is a function that returns the posting price, and $||$ is the concatenate operator. The two conditions of the optimization problem are interpreted as follows. First, after receiving a new bid, b_i, the arbiter will only change $p_t(d)$ if it makes more revenue by doing so: b_i must cause a change in $p_t(d)$. The second condition says that the price

must be changed in the direction that benefits the buyer: b_i^* lowers the price, increasing the buyer's utility. The buyer cannot solve the above optimization problem because it does not know what bids are part of the epoch. As a consequence, buyers no longer know what bids guarantee to drive prices down.

3.3 Analysis

The larger E, the larger the amount of information a buyer needs to learn to solve the optimization problem. For that reason, we say that E determines the degree of protection. Unfortunately, higher protection does not come for free.

Claim 1: (Protection-Revenue Tradeoff) In an online posting price data market, where posting prices are chosen based on epochs, the larger the epoch size, the lower the optimal revenue. Consider a vector of bids, $\vec{b}$, and the optimal revenue $r(\vec{b}) = k * b_k$, where b_k is the k-th largest bid in $\vec{b}$ and k the position of b_k in the sorted vector $\vec{b}$. k is computed based on equation 2. We want to show that if we partition $\vec{b}$ into two partitions, $\vec{b_1}$ and $\vec{b_2}$, then $r(\vec{b}) \leq r(\vec{b_1}) + r(\vec{b_2})$. To show that, consider we construct $\vec{b_1}$ by copying all bids from $\vec{b}$ except for one, b_c. $\vec{b_2}$ only contains b_c. Consider two scenarios. First, if $b_c < b_k$, then $r(\vec{b}) = r(\vec{b_1})$, so $r(\vec{b}) \leq r(\vec{b_1}) + r(\vec{b_2})$. If $b_c \geq b_k$, then $r(\vec{b_1}) = r(\vec{b}) - b_k$, and $r(\vec{b_2}) = b_c$. Since $b_c \geq b_k$, then $r(\vec{b}) \leq r(\vec{b_1}) + r(\vec{b_2})$. The result generalizes to more partitions by induction. This shows that more partitions lead to higher revenue. Higher E means fewer partitions and therefore lower revenue.

The buyer social surplus, defined as the total utility of buyers, depends on the allocation decisions for each buyer, X_i, as well as the difference between v_i and $p_t(d)$. The maximum social surplus is achieved when $p_t(d) = b_k = 0$ because any $b_i \geq 0$ will be allocated the dataset and because this maximizes the difference between v_i and $p_t(d)$, i.e., the buyers' utility. Then, it is clear that as b_k grows, social surplus decreases. However, there is no clear relationship between changes in E and changes in b_k.

4 TIME-SHIELD PROTECTION

4.1 Preliminaries

Buyer's strategy. We assume that a buyer can bid at most once per time period, $t \in [0, \ldots)$, and this is enforced by the arbiter. A buyer with patience τ_i can bid a total of $T_i = \tau_i - t$ times, with t indicating the current period. After the deadline, the buyer's utility becomes zero, so the buyer wants to win the allocation before the deadline. The buyer has T_i allocation opportunities to win the dataset.

A buyer who wins an allocation within its deadline maximizes its utility when the gap between its private valuation, v_i, and the price, $p_t(d)$ is largest. Because buyers know the arbiter adjusts $p_t(d)$ based on past bids, strategic buyers may design a sequence of $T_i - 1$ bids aimed to drive prices down. Strategic buyers bid low with the intention of influencing future prices so that $p_{\tau_i - 1}(d) \leq p_t(d)$, hence increasing their utility.

Introduction to Waiting Protection. In an ideal protection scenario, when a strategic buyer submits a *strategic bid* ($b_i < v_i$), the arbiter prevents them harming the market performance by letting that strategic buyer bid exactly only one more time before their deadline, hence, giving them only one more chance to win the allocation. This can be achieved by making strategic buyers wait for a *wait-period* w_i. When $w_i = T_i - 1$, the buyer's patience is exhausted. This wait-period forces strategic buyers to participate in what equates a single-bid round where they cannot strategize anymore without risking losing utility.

In practice, the arbiter does not know the buyer's deadline τ_i so it cannot set $w_i = T_i - t$. If set too short, buyers will ignore the wait-period penalty and continue strategizing. If set too long, $w_i > T_i$, buyers utility becomes 0, and the market extracts no revenue because there is no payment. Furthermore, if $b_i \leq v_i$, the arbiter cannot tell if b_i is strategic or the buyer's valuation is low.

4.2 The Time-Shield Technique

The goal of the Time-Shield technique is to choose a w_i that disincentivizes strategic buyers (**Claim 2**) but without harming truthful but losing buyers' utility (**Claim 3**).

The insight behind the technique is to choose $w_i > 0$ to disincentivize strategic behavior but in a way that guarantees no harm is caused to truthful buyers that lose their allocation, $b_i = v_i < p_t(d)$. In other words, choosing w_i so the truthful buyer cannot lose utility before $t + w_i$. The arbiter can do this because it knows $a(\hat{b})$, the past bids, and the losing bid, so it can calculate the minimum time at which b_i would become competitive by simulating a future sequence of bids that would make $b_i \geq p_t(d)$. In other words, w_i is set up so the buyer would not have been able to win the allocation earlier no matter if the original losing bid was truthful or not. We demonstrate both claims next.

Claim 2: $w_i > 0$ disincentivizes strategic behavior. The buyer's utility depends on winning an allocation within its deadline and on the magnitude of the difference between the posting price and the buyer's private valuation. A buyer needs to win only one time within the deadline. Let us denote the probability of winning any one allocation as $k, 0 > k > 1$. The buyer's probability of winning an allocation is the same for any time period, and the buyer's utility is the same whether they win the allocation at the first or last time period. Then, a buyer prefers to have more opportunities for winning an allocation. When $w_i = 0$, a strategic buyer has exactly T_i allocation opportunities. When a wait-period is set, $w_i > 0$, the number of opportunities reduces to $T_i - w_i$. Hence, waiting disincentivizes strategic bids because buyers prefer having more allocation opportunities.

There is another disincentive property of wait-period. While without wait-period buyers know exactly their allocation opportunities, T_i, when $w_i > 0$, their opportunities are strictly reduced, but they *do not know how much* because w_i is private to the arbiter. In particular, the buyer does not know if $w_i > T_i$, which would yield $u_i = 0$. This additional uncertainty plays an important role in neutralizing strategic behavior as we demonstrate empirically in the evaluation section.

Claim 3: Time-Shield does not reduce truthful buyers' utility. We consider two scenarios, $b_i \geq p_t(d)$ (winning, $X_i = 1$), and $b_i < p_t(d)$ (losing, $X_i = 0$). When $X_i = 1$, $w_i = 0$, so utility is not affected by Time-Shield. The interesting case is when $X_i = 0$. In this case, $w_i > 0$. u_i depends on w_i, $p_t(d)$, and X_i:

- if $w_i > T_i$, then $u_i = 0$, because the buyer will not bid after the deadline. In this case, the buyer's utility is harmed only if $\exists t' \in [t, T_i] \mid p_{t'}(d) < b_i$, i.e., if there was an opportunity for the buyer to win the allocation in the range $[t, T_i]$. In other words, the buyer could have won the allocation but w_i reduced its chances. If the above condition is false, then, despite missing the deadline, the buyer would have never won the allocation by bidding b_i, so its utility would not be harmed.
- if $w_i < T_i$, and $\exists t' \in [t, w_i] \mid p_{t'}(d) < b_i$, then this reduces the buyer's allocation opportunities. However, in this case, the buyer has more opportunities to bid in $[w_i, T_i]$, so it can still win the allocation and therefore we cannot determine its u_i yet.

Then, as long as $\nexists t' \in [t, w_i] \mid p_{t'}(d) < b_i$, Time-Shield does not harm the buyer's utility. By design, Time-Shield chooses w_i so that there cannot be prices lower than the losing bid within the wait-period. Therefore, Time-Shield does not affect buyer's utility. We describe how to choose w_i in Section 6 because it requires details of the pricing algorithm.

5 UNCERTAINTY-SHIELD PROTECTION

In this section, we describe **Uncertainty-Shield**, a technique to protect data markets against two challenges: i) buyers who guess p based on the wait time, w_i, as determined by Time-Shield; and ii) boundedly-rational behavior.

Guessing prices based on w_i. According to Time-Shield, a losing buyer will wait w_i before bidding again. Buyers will learn w_i once they are allowed to bid again and they know w_i is set based on $a(\vec{b})$: this leaks information about the price at the time of the bid, $p_{t-1}(d)$.

Boundedly-Rational Behavior. Consider a buyer who is about to bid for a dataset truthfully, $b_i = v_i$. Just before submitting its bid, the buyer learns about the price at which such a dataset was sold moments ago, p_{t-1}. Armed with the price information and knowing the arbiter sets prices based on past bids, the buyer may guess that the current price will be *similar* to the leaked price, $p_t(d) \approx p_{t-1}(d)$. If the buyer was originally bidding truthfully, and above the price, $b_i \geq p_t(d)$, should the buyer change its bid in light of the new information? In particular, should the buyer lower its bid so that: $b_i \approx p_{t-1}(d)$?

The answer is no because the buyer's utility is determined by its private valuation and the posting price that the buyer pays, and not the actual bid. Lowering the bid only risks losing the allocation when $b_i < p_t(d)$ and hence harming the buyer's utility. Still, we observe empirically that some buyers change their bid. Unfortunately, these lower bids will contribute to future posting prices, hence harming the market revenue.

Uncertainty-Shield. The crux of the problem is that buyers assume $p_t(d) \approx p_{t-1}(d)$ because they know prices are set based on past bids, and they do not realize that *what they bid is not what they pay*[5]. Uncertainty-Shield protects against this behavior by adding noise to the posting price $p_t(d)$ after every epoch. This increased price uncertainty tames buyers' non-rational behavior and protects against buyers guessing the price based on w_i because it breaks the link between price and past bids.

The protective effect of adding noise to the posting price is at odds with the *update algorithm's* goal. For example, too much noise will effectively randomize prices completely, hence losing the learning effect from the algorithm and harming revenue as a consequence. The main challenge of Uncertainty-Shield is to include this noise to endow the market with this protection while maintaining performance guarantees on the revenue raised by the market. We explain how to achieve this in the next section.

[5]Previous research has observed that the additional cognitive effort of understanding that 'what you bid is not what you pay' makes certain auction formats harder to understand to participants [38].

6 PUTTING ALL TOGETHER: A PRICING ALGORITHM FOR DATA MARKETS

In this section, we present a pricing algorithm that combines Epoch-Shield, Time-Shield, and Uncertainty-Shield to protect data markets from strategic buyers (Section 6.2). We then discuss an alternative design that uses differential privacy (Section 6.3). We start by summarizing the requirements of the algorithm.

6.1 Summary of Requirements

R1. Online pricing. Given an incoming stream of bids, $\vec{b}$, whose distribution is unknown and changing, the pricing algorithm must choose and adapt $p_t(d)$ in order to maximize revenue using Epoch-Shield to protect against low bids.

R2. Making losing buyers wait. The pricing algorithm must compute a wait-period, w, for every losing buyer to exhaust their patience and disincentivize strategic behavior. This is the Time-Shield protection technique.

R3. Controlling Uncertainty. The price, $p_t(d)$ must be set with sufficient randomness that buyers who learn about leaks are disincentivized to adjust their bids. This is the Uncertainty-Shield protection technique.

6.2 The Pricing Algorithm

The overarching goal of the algorithm is to choose a posting price that maximizes revenue before knowing the future bids but using knowledge of past bids. This problem is an online decision making process [12], where the decision is to choose an *expert*, and experts correspond to candidate posting prices. To select an expert in an online manner the algorithm maintains a list of candidate posting prices and assigns *weights* that are updated every epoch, after observing the performance of each expert/posting price.

We choose to update the weights using the multiplicative weights rule, which has the following desirable features:

- It delivers strong performance guarantees. The revenue obtained using MW is the one obtained by the best expert in hindsight [6].
- It permits us to compute the Wait-Period required by Time-Shield because we understand how price changes given a stream of bids.
- It models weights as a probability distribution and chooses a weight according to that distribution. This naturally incorporates the randomization demanded by Uncertainty-Shield while maintaining performance guarantees.

6.2.1 Allocation and Update Logic. At a high level, the algorithm must: i) execute the allocation function for each incoming bid based

Algorithm 1: Data Market Pricing Algorithm

```
   input : b⃗, input bid stream,
           P, collection of posting
   price candidates,
           E, epoch size,
   // Initialize
      multiplicative weights
   // We indicate p_t(d) as p_t
1  MW ← init(P);
2  p_t ← MW.draw_price();
3  epoch ← [];
4  for b_i ∈ b⃗ do
5      epoch.add(b_i);
       // Process bid
6      if b_i ≥ p_t then
7          x_i ← 1;
8          handle_payment();
9      else if b_i < p_t then
10         x_i ← 0;
11         w_i ← com-
             pute_wait_period(MW,
             b_i);
12     p_t ← update_price();
       // Update price when epoch
          is complete
13 def update_price(epoch, E, p_t,
     MW):
14     if length(epoch) != E then
15         return;
16     opt_r ← optimal(epoch);
17     revenue ← r(epoch, p_t);
18     for χ ∈ MW.experts do
19         alt_r ← r(epoch, χ.p);
20         c_i ← (revenue - alt_r) /
             opt_r;
21         if c_i ≥ 0 then
22             χ.w ← w(1 − ε)^{c_i}
23         else if c_i < 0 then
24             χ.w ← w(1 + ε)^{−c_i}
25     p_t ← MW.draw_price();
26     return p_t;
```

on the current posting price, $p_t(d)$; ii) group incoming bids per epoch, and update $p_t(d)$ after every epoch is done. The algorithm chooses a price that maximizes revenue before seeing future bids based on past bids while implementing the Epoch- (requirement **R1**), Uncertainty- (**R3**), and Time-Shield (**R3**) techniques.

Initializing algorithm. Algorithm 1 expects a stream of bids, $\vec{b}$, and is configured with a set of posting prices, P[6] that we assume fixed for the sake of presentation, and an epoch size E. This is shown in the input parameters of Algorithm 1. The algorithm then initializes the experts' weights for each $p = 1\ \forall p \in P$, line 1. Each expert is referred to as χ, its weight as $\chi.w$ and its value, the posting price, as $\chi.p$. After initialization, an initial posting price, $p_t(d)$ is chosen when processing the current epoch.

Processing incoming bids. For each incoming bid b_i (line 4), the algorithm includes the bid in the current epoch, and then compares the bid with p_t[7] (line 6) and makes an allocation decision. When the allocation decision is positive, the next step is to handle the payment (line 8). If the allocation is negative, the Time-Shield technique kicks in to compute the wait period, w_i (as explained in Section 6.2.2). Note that after receiving a bid, the algorithm can immediately decide whether to allocate and handle the next steps because a posting price has been chosen a priori: buyers do not need to wait, and hence their utility is not harmed.

Updating posting prices. We use MW to update the experts' weights (i.e., the posting prices) adaptively, after every epoch. After processing each bid, the algorithm checks when an epoch is completed. An epoch is completed when the number of bids it contains equals E, (see line 14). If the epoch is finished, the algorithm computes the *cost* of the current p_t and uses it to update the weights of all other experts before choosing the next p_t. This addresses **R1**.

[6]each posting price corresponds to an expert in MW

[7]Note we use the notation p_t in the algorithm because the dataset it refers to, d, is implicit in the execution context.

The cost is measured as the relative revenue difference. This is the cost that each expert would have incurred had it been chosen (line 19), compared to the chosen p_t (line 17), and normalized by the optimal posting price, which is computed in line 16 after all bids within the epoch are known. With a cost per expert, their weights are updated following the MW rule, lines 21 to 24. Once all expert weights have been updated, the algorithm chooses a new p_t (line 25), which becomes the posting price for the incoming epoch.

Choosing the next posting price. The function draw() (line 25) does not select p_t deterministically. Instead, it follows the multiplicative weights rule. It interprets the experts' weights as a probability distribution and samples p_t based on that distribution, i.e., experts with higher weights are more likely to be selected. This, in effect, adds the uncertainty required by Uncertainty-Shield (addressing **R3**) while maintaining strong performance guarantees. In particular, the multiplicative weights rule ensures that the expected penalty, in our case in revenue, is not much worse than that of the best expert in hindsight. The proof is in the original work [6].

6.2.2 compute_wait_period and MW. When $X_i(b_i, p_t(d)) = 0$, the algorithm computes the wait-period, w_i, that indicates the number of time periods the buyer will need to wait before bidding again (addressing **R2**). w_i indicates the time needed before b_i would become *competitive* and is computed so that truthful losing buyers do not lose utility as seen in Section 4.2.

In the context of the MW-based pricing algorithm, a bid is competitive if it is close to the most likely posting price, i.e., the posting price with the highest weight. The algorithm simulates the sequence of future bids at which b_i would become competitive, and measures the number of bids necessary to achieve that. Although in reality a bid may never become competitive, the sequence of bids chosen by the algorithm are the best bet for the buyer. We consider two strategies to replay future bids and hence to implement compute_wait_period():

- **Bound.** The algorithm assumes that the sequence of future bids contains bids with the minimum possible value, guaranteeing that the number of iterations until b_i converges is minimum, i.e., the experts' weights change faster because the lowest bids incur the highest costs. w_i, computed this way, is the earliest time at which b_i can win any allocation, so even if the buyer were to bid earlier, it would not win.
- **Stable.** In this strategy, the algorithm assumes the incoming bids have all values equal to b_i. This strategy gives a more conservative estimate of when the bid may become competitive.

The goal of **Time-Shield** is to guarantee that if b_i was truthful, the buyer would not lose utility when compared to a non-waiting mechanism; if b_i never becomes competitive, then waiting does not change the buyer's utility. For this reason, both the Bound and Stable strategies are optimistic on the side of the buyer. In our market model, buyers can bid once per time period $t \in [0, \ldots)$. Then, given the incoming rate of bids, which depends on the number of buyers at a given time, the algorithm computes the time periods within which the buyer cannot bid; the number of time periods corresponds to w_i.

6.3 The Differential Privacy Angle

We discuss an alternative implementation of Epoch-Shield and Uncertainty-Shield based on differential privacy [24]. A low bid is harmful when it has a significant effect on the revenue raised, which in turn depends on the price at which the dataset is sold. The role of an epoch is to *smooth out* the influence of single bids. Given a bid, b_i, this intuition can be expressed as follows, $pr(\vec{b})/pr(\vec{b}||b_i) < \epsilon$, which corresponds to the definition of differential privacy (DP). The numerator and denominator differ in one value, b_i, and the ratio of the prices must be lower than a parameter, ϵ, which in the context of DP corresponds to the privacy, and in the context of Epoch-Shield corresponds to the degree of protection: the lower ϵ is, the more protected the market is against low bids.

The arbiter would then collect bids from buyers and compute a differentially private version of the price; one that by definition is not affected much by any single bid. This could be done, for example, using the Laplace mechanism [24]. The arbiter would then calculate the price as $a() + Y$, where $Y \sim Lap(\lambda)$. *Lap* is the Laplacian distribution, and $\lambda = S(a)/\epsilon$. The *protection* (privacy) parameter, ϵ, and the *sensitivity* of the allocation function are used to parameterize the distribution from where noise is drawn. Finally, the sensitivity is defined as $S(a) = \max |a(\vec{b_1}) - a(\vec{b_2})|_1$. When a is defined as in Equation 2, then the value is $S(a) = \max(b) - \min(b)$, i.e., the difference between the highest and lowest possible bids.

Alternative implementations. We believe the MW-based algorithm is conceptually and practically simpler to implement than a market based on the Laplace-mechanism because the MW-based algorithm does not need knowledge about the highest and lowest bids, and can incorporate the Time-Shield technique in a straightforward way, as explained above.

Note on related work. Differential privacy has been used to endow an auction mechanism with protection against strategic buyers [2], although in an auction format not compatible with data markets. The connection between differential privacy and mechanism design has been made before [30, 46].

7 EVALUATION

In this section, we present the results of a user study that we conducted to demonstrate the need for the protection techniques we have presented and their effectiveness in mitigating the effect of strategic buyers (Section 7.1). Later, in Section 7.2 we present the results of extensive simulations that describe the performance characteristics of our techniques.

7.1 Are Protection Techniques Effective?

We answer the following 5 research questions using a user study. We first show the bid distribution generated by participants of a data market.

- **RQ1: Do buyers bid truthfully in the data market?, i.e., will buyers strategize?**

We demonstrate the boundedly-rational behavior of participants and hence the need for Uncertainty-Shield (**RQ2**) and then we show the protection effect of Uncertainty-Shield in **RQ3**:

- **RQ2: Do price leakages drive buyers' bids down when they know prices are set based on past bids?**
- **RQ3: Does Uncertainty-Shield help to preserve bids despite leakages?**

We demonstrate that buyers will strategize over time and hence the need for Time-Shield (**RQ4**) and then we demonstrate the effectiveness of Time-Shield in **RQ5**:

- **RQ4: Do buyers act strategically when they bid over several rounds?, i.e., will buyers strategize over time?**
- **RQ5: Does Time-Shield make buyers bid truthfully even in multi-round sessions?**

To answer these questions, we design and conduct an IRB-approved user study involving human subjects, which we describe next.

Setting. The study describes a market environment where buyers (i.e., the participants) work as data traders for a company. Participants' goal is to obtain the datasets their company requests. As described in this paper, their utility function is the difference between their company's valuation and what they pay for the dataset. In each scenario, we tell participants how much their company values a dataset, v_i. Then, participants bid to obtain the dataset. The valid bid range is $[0, 2 * v_i]$, so participants can bid higher than their valuation (non-rational), their valuation (truthful), or lower than their valuation. Participants use a slider to choose the bid.

Money and real stakes. Participants do not bid with real money. This introduces a potential threat. Since money is not real, they may value it less and that would lead to unnecessarily high bids or worse, random behavior because participants are not vested in the task at hand. However, none of these threats show up in our study results. The results demonstrate that participants indeed aim to maximize their utility i.e., they aim to get the best deal for their company as described in the scenario above. For this reason, and despite the threat of validity, we believe the user study motivates well the challenges and demonstrates the value of the contributions made in the paper.

Recruitment and data collection. We recruited 53 participants online out of which 50 completed the study and were compensated. Each participant signs a consent form that clearly describes the content and purpose of the study. We used Prolific [57] to recruit participants for 3 reasons.

First, Prolific's participants have a high approval rate, measured as the ratio between successfully completed studies and the total returned studies, which indicates participants are engaged and committed to complete the study. Second, Prolific is based in Europe and hence subject to GDPR [29], which means participants can request their data to be removed. Third, Prolific uses a wide variety of demographic variables that we use to select the sample of target participants to reduce the threats to the external validity of our study, as explained later.

We did not collect from participants any identifier or other PII information. We did not collect their IP address or other information that may disclose their location. We only collect participants' answers to the study questions, which we store in a secure enclave and we analyze to present aggregated results as part of this study.

Methods. To eliminate subject-specific differences, we conduct a within-subjects study: each participant answers all questions. We randomize question order to avoid ordering effects. The study starts with the answer of **RQ1** as a baseline over which we introduce

interventions to understand the effect of leakages, price randomization, multiple-round bids, and the Time-Shield technique. We measure each intervention's effect by comparing the results of different questions to each other as we indicate in the next section.

When analyzing results, we use the statistical testing framework. We formulate the null and alternative hypothesis, measure aggregate statistics, and use a statistical test to see if we can reject the null hypothesis. When choosing a test we take into consideration two characteristics of our data and study design, respectively. First, the collected data does not follow a normal distribution. We reject this null hypothesis using the Shapiro's [14], and D'Agostino and Pearson's test [20]. This characteristic calls for a nonparametric test. Because our study is within-subjects, when comparing questions, we are comparing paired samples. Therefore, we use the Wilcoxon signed-rank test [56], and indicate any variations.

7.1.1 Research Questions and Results. **RQ1: Do buyers bid truthfully in the data market?** We consider two scenarios. In the first, the company values the dataset at 500, and in the second at 1500. We ask participants to submit their bids in each case. The results in Table 1 show that both the average and median bids are close to the truthful bid, and the standard deviation is not zero. The complete distribution of bids is shown in Figures 2a and 2b labeled as No-leak. The figures show that the distribution of bids is concentrated near the truthful bid, but some participants bid below, and some bid above. We use a 1-sample Wilcoxon test to test the alternative hypothesis that the median of the sample is different than the median of the distribution, and obtain $p \geq 0.3$ in both price cases: we cannot reject the null hypothesis, so we conclude the median of the sample is the median of the distribution. The results show that not every participant bids truthfully, although the distribution of participants bids nearly truthfully. This evidence is consistent with extensive research in behavioral economics that demonstrates that participants not always act rationally [39]. We do not attempt to characterize why this behavior takes place but based on our pilot tests, it is often due to ignorance, lack of attention, or unstated assumptions participants make. We accept this behavior as the norm in the wild and refer to the distribution of bids centered around the truthful bid as *near-truthful*.

	Mean	Std	Median	p-value
500	456	81.66	450	0.35
1500	1368	191.24	1350	0.32

Table 1: Statistics for RQ1

RQ2: Do price leakages drive buyers' bids down when they know prices are set based on past bids? We put the participants in a position where before bidding they learn that the arbiter sets prices based on past bids and they get access to the latest price set by the arbiter. We then ask them to bid. Figures 2a and 2b show the distribution of bids when participants do not know about the leak (No-leak) and when they learn about the leak and know prices are based on past bids, labeled Past. What the distributions of bids show visually, that leakages cause a drop in bids, is confirmed by the statistical test: we reject the null hypothesis and *conclude that participants do not behave rationally when they learn about leakages and understand that prices are set based on previous bids.* This is the motivation for the Uncertaintiy-Shield protection technique.

RQ3: Does Uncertainty-Shield preserve prices despite leakages? This is the same situation as before, but we tell participants the arbiter sets prices randomly. The results are labeled in Figures 2a and 2b with Random. The distribution of bids, in this case, is flatter than in the case of No-leak and Past. Randomizing prices does not preclude all participants from dropping their bids ($p < 0.01$), but it ameliorates the problem significantly, $p < 0.01$. We conclude that *randomizing prices is an effective way of ameliorating the effects of non-rational behavior.* This demonstrates the effectiveness of Uncertainty-Shield.

RQ4: Do buyers act strategically when they bid over several rounds? In this case, participants can bid once per hour, and they are given several hours to bid. If they bid and lose, they can bid again as long as they have chances left. We ask them to write a bidding plan with the sequence of bids they would submit. In this setting, buyers act strategically by placing lower bids initially and bidding near truthfully in their last chance. The results are presented in Figure 2c, where the dataset price is 2000. The label NW-p25 shows the 25th percentile of bids, NW-median the median, and NW-p75 the 75th percentile. We conducted a second experiment with price 600 and observed similar results. These results *demonstrate that buyers will act strategically to maximize their utility.* This is the motivation for the Time-Shield protection technique.

RQ5: Does Time-Shield make buyers bid truthfully even in multiple-round sessions? This situation is the same as above. However, we tell participants that if they bid and lose they may need to wait an amount of time proportional to the difference between the bid they made and the price set by the arbiter, i.e., we use Time-Shield. We explain that the consequence of this is they may miss opportunities for bidding. The 25th, median and 75th percentiles are shown with W-p25, W-median, and W-p75, respectively, in Figure 2c. We differentiate the last bid (hours 4 and 3 respectively) from the previous ones. Using Time-Shield drives bids up, ameliorating the strategic behavior of participants, who know they may lose the opportunity to acquire the dataset. It does not fully make bids near-truthful, except for the last bid. The difference in bids is statistically significant ($p \leq 0.01$) in all hours except for the last one. The last hour is the last chance buyers have to acquire the dataset, with and without Time-Shield, so in this case, we observe, as expected, that bids are near-truthful in both scenarios. *The results show that Time-Shield helps disincentivize buyers from strategizing, therefore driving bids up.*

7.1.2 Threats to validity and limitations. **External Validity.** To reduce spurious behavior from participants, we recruited those who were familiar with trading environments, financial information, who are currently part of the active population, whose minimum degree of education is a graduate degree (Ma/MSc/MPhil/other), and who consider their industry to be described as *Finance*.

Internal validity. We designed the study to minimize threats to its internal validity:

Language. We recruited participants from the US and the UK whose first language is English. We conducted a pilot test with 6 volunteers who agreed to explain their interpretation of the questions aloud.

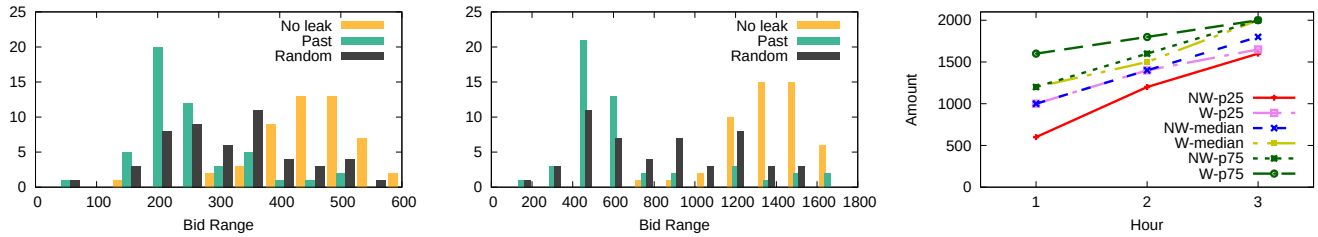

Figure 2: Left: Distribution of bids without leaks (No leak), when participants learn about a leak and know prices are set based on past bids (Past), and when prices are set randomly (Random). Center: Same as left but for price 1500. Right: Bids with (W) and without (NW) Time-Shield for price 2000. Showing 25th (p-25), 50th (p-median), and 75th (p-75) percentiles.

This helped us hone the language to avoid ambiguity and make it easier to understand.

Market understanding. There is a chance that participants do not understand how the data market works or they forget essential details while participating. After explaining the instructions, we included a question to test the participants' knowledge. When they get the answer wrong (44% got the answer wrong the first time), we offer an explanation and clarification of why that is the case. We found that including examples helped participants to grasp the mechanics quickly, so we included an example with the study instructions, and we added it to a refresher that they had available throughout the survey. In this way, participants could always go back and read the instructions.

Price effect. The price magnitude may affect how participants bid. To control for this, we include the above questions for different prices that differ in one order of magnitude, and we randomize the order in which we present different prices to different participants.

Learning effect. When participants learn about leakages (**RQ2 and RQ3**), they are requested to provide their bid under two scenarios: arbiter sets prices based on past bids and randomly. To control for learning effects, we randomize the order of the questions.

7.2 Data Market Simulation

In this section, we study the performance implications of the protection techniques using simulation data [49]. We organize the section around 3 research questions:

- **RQ6: How does Epoch-Shield affect performance?**
- **RQ7: How does Uncertainty-Shield affect performance?**
- **RQ8: How does Time-Shield affect performance?**

When measuring market performance, we use revenue and social surplus: the aggregated utility across all *buyers*.

We complement these results with a comparison of pricing algorithms and an exploration of strategic behavior. We start by describing the data generation process.

7.2.1 Data Generation and Strategic Buyers. We run simulations over time series of bids. Each point in the time series represents the buyer and its bid. We generate time series according to an autoregressive model [5], which is common in econometrics. Because each time series is generated randomly, we generate 100 random time series of 250 points each for each experiment and present aggregate results. In particular, we show the 1, 25, 50, 75, and 99 percentile in boxplots. All revenue and social surplus results are normalized to the maximum value; this is because what matters is the relative difference between configurations and not the absolute value, which is an artifact of the range of bids chosen.

The time series generation process is governed by a parameter, *AR*, that controls how much the value of a point in the time series depends on previous values. We tried many different parameters[8] and measured their effect on the normalized revenue for the optimal posting price algorithm, Opt, and our pricing algorithm, MW, as shown in Fig. 3a. We found out that the performance is not too sensitive to this parameter. In the remainder of the evaluation, we present results corresponding to $AR = 0.1$ because the results for other values of *AR* are similar and do not show new insights. All our code will be open sourced, so other researchers may try different parametrizations of the generation process.

Generating strategic buyers. To simulate strategic buyers, we transform the time series using a function that takes a triple $< PCT, \beta, H >$. The triple describes the strategic buyers. *PCT* determines the ratio of buyers that act strategically. For each strategic buyer, H is the horizon over which it bids, i.e., it corresponds to T_i. Finally, $\beta \in [0, 1]$ multiplies the buyer's true valuation, v_i, lowering the bid; we use β to determine the value of the strategic bid, with lower values leading to more aggressively low bids. $PCT = 0$ corresponds to the ideal scenario where all buyers are truthful.

7.2.2 RQ6: How does Epoch-Shield affect revenue and social surplus? We want to study the protective effect of epoch size as a function of PCT. In this experiment, strategic buyers bid the minimum over a fixed horizon, H, unknown to the market arbiter.

Figures 3b and 3c show normalized revenue and social surplus, respectively, as PCT (x axis) grows and for 5 different values of E. In Fig. 3b when PCT= 0, corresponding to the truthful scenario, we see how larger E leads to lower revenue, as expected. However, the key is that, as PCT grows, smaller values of E lead to a revenue collapse, while higher protection leads to the market maintaining the performance. In particular, we see differences of 10x in this experiment between different epoch sizes. The most interesting insight of social surplus is that the performance for large epochs remains similar across PCT values and does not drop with higher protection. **The protection offered by larger epoch sizes compensates for the loss of revenue caused by strategic buyers while maintaining the social surplus.**

[8] We tried the following tuples (ar, σ) = (0.1, 0.01), (0.5, 0.01), (0.9, 0.01), (0.999, 0.01)

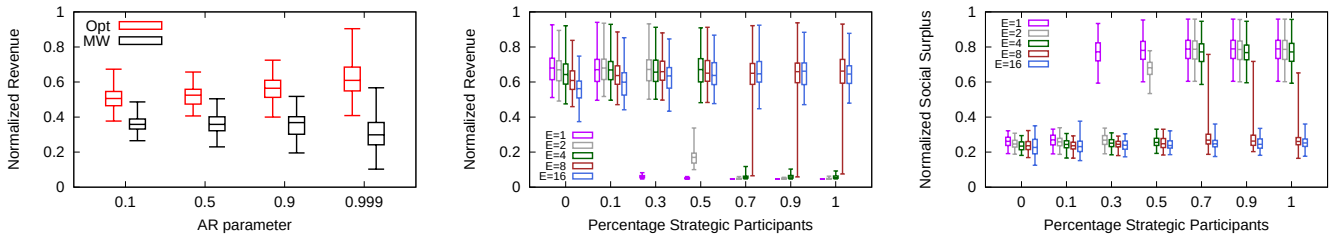

Figure 3: Left: Normalized revenue of Opt and MW under different AR parameterizations. Center: Normalized revenue of different epoch sizes as PCT increases. Right: Normalized social surplus of different epoch sizes as PCT increases.

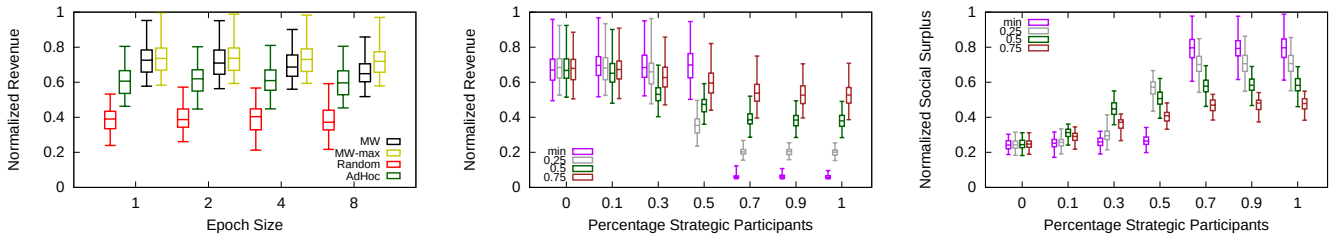

Figure 4: Left: Normalized revenue of MW versus MW-max, Random, and AdHoc for different epoch sizes. Center: Normalized revenue for different β as PCT increases. Right: Normalized social surplus for different β as PCT increases.

7.2.3 RQ7: How does Uncertainty-Shield affect revenue? Uncertainty-Shield tempers participant's boundedly-rational behavior as shown in **RQ3**. But by adding noise it interferes with the goal of raising revenue. We use the multiplicative weights rule to select a randomized price, hence guaranteeing the revenue performance according to the algorithm's proof [6]. Here, we empirically demonstrate the effect of adding noise using 3 baselines and our approach and show the results in Fig. 4a (for a truthful scenario with β and H configured as in the previous experiment).

MW, AdHoc, and Random all add noise to the price, while MW-Max deterministically selects the price with the highest weight. While MW-Max achieves higher revenue than the other baselines it does not implement Uncertainty-Shield so when deployed in the wild, it will suffer from the problems we demonstrated in RQ3. Of the mechanisms that include noise and hence offer protection, Random chooses prices completely at random, breaking any link between price and past bids. The consequence is very poor performance. AdHoc selects a price by sampling the neighborhood of the price with the highest weight. Although this mechanism performs much better than Random, it ignores the actual weights and hence does not provide any performance guarantees. Finally, our implementation, MW performs best among the randomized baselines and provides guarantees on the performance, demonstrating it is an appropriate way of implementing Uncertainty-Shield.

We conclude **the overhead of Uncertainty-Shield is small and justified by its effectiveness in avoiding the harmful effects of non-rational behavior.**

7.2.4 RQ8: How does Time-Shield affect revenue and social surplus? In this experiment, we fix H as in the previous experiments and choose $E = 8$, which has shown an adequate level of protection to the market as previous experiments have shown. With this setting, we seek to understand the effect that Time-Shield has on the data market performance.

Fig. 4b and 4c show the normalized revenue and social surplus of the market when β, the parameter used to set the strategic bid, changes. As PCT grows, revenue lowers, as expected. When strategic buyers bid the lowest possible, this harms the market the most (this setting is referred to as min in the graphs). As β grows, strategic bids resemble the truthful bid that corresponds to the original private valuation, v_i. Higher values of β lead to higher revenue, with large differences as can be seen when PCT is ≥ 0.7. The user study of the previous section demonstrated that with a waiting technique (Time-Shield), strategic buyers drive their bids up—equivalent to higher β. For that reason, we conclude that **Time-Shield prevents the harmful effect on market revenue caused by buyers that strategize over time and therefore improves market performance**. Besides, we know that in the presence of Time-Shield, specific strategic buyers stop being strategic. This leads to higher revenue as the figure shows—see lower values of PCT.

7.3 Pricing Algorithms and Strategic Bid Study

We complement the simulation results with a performance comparison of different algorithms, demonstrating that our choice of algorithm outperforms other baselines (Section 7.3.1) and finish with a study of the effects of different parameters that configure strategic bids in Section 7.3.2.

7.3.1 Why not just using average or median? Update algorithms such as average (avg), and median (p50), are susceptible to strategic buyers, who know that a low bid always affects the price chosen by the algorithm in a direction that is profitable for the buyer. In addition, the avg and p50 performance are worse than our pricing algorithm, MW, which builds on top of multiplicative weights and

therefore can adapt to unknown bid distributions better. Fig. 5a shows the normalized revenue of the different update algorithms. As PCT increases, the performance of avg and p50 drops dramatically, affected by the low bids. In contrast, the performance of MW remains close to the optimal, Opt, throughout the experiment.

7.3.2 How do strategic bids affect market performance? In this experiment, we seek to understand the effect of H (labeled as horizon in the graphs) as well as the magnitude of the strategic bid, β, on market performance. We produce heatmaps that show the normalized revenue when these parameters change. We first conducted a baseline experiment where PCT is 0.1. With so few strategic buyers, we expect the normalized revenue not to drop much and the baseline confirms this is the case.

Fig. 5b and Fig. 5c show heatmaps when PCT is 0.5 and 0.9 percent respectively. The larger the horizon and lower the strategic bid, the worse the revenue. For 50% and 100%, the normalized revenue is as low as 0.1 and 0.2. Importantly, higher β leads to higher revenues even when horizons are large—this reinforces why Time-Shield is an effective market protection.

8 EX-POST ALGORITHM DISCUSSION

8.1 Ex-Post Bidding for Experience Goods

We have considered so far cases where buyers know v_i. When buyers do not know v_i, they cannot bid without risking to pay more than what the dataset is worth to them. Buyers may not know how to value a dataset before using it, for example, when engaging in exploratory tasks.

The market could allocate datasets to buyers and then accept payments after buyers have used the datasets and know how to value them. Arrow's information paradox [7] means that buyers do not have an incentive to report the true valuation after accessing the dataset. We discuss how the Time-Shield technique could be adapted to this ex-post scenario.

8.2 Truthful Ex-Post Pricing

Many buyers buy more than one dataset over time, $d_1, ..., d_N$, and they are interested in maximizing their total utility, $\sum_n^N u_i(d_n)$. Ideally, the revenue raised by the market is the same in the ex-ante (i.e., when buyers know their v_i) and ex-post scenario. The key idea to elicit truthful payments from buyers is to penalize them with Time-Shield *the next time* they bid for a dataset.

Concretely, after a buyer sends a payment for a dataset it has already used, the arbiter compares the payment, $P(d_1)$, with the posting price associated with d_1 at the time t of the dataset allocation, $p_a = p_t(d_1)$. If $P(d_1) \geq p_a$, then this corresponds to a case where the buyer did not cause any revenue loss, so the arbiter simply collects p_a. When $P(d_1) < p_a$, then the arbiter collects $P(d_1)$ and then uses the **Time-Shield** technique in the following way. It computes w_i based on the difference between $p_a - P(d_1)$ and then makes the buyer wait *the next time it bids* for another dataset, d_2. Because buyers seek to maximize their utility over time, the prospect of having to wait later is a deterrent to bid untruthfully.

What about truthful losing buyers. Buyers may bid truthfully and still lose, in which case they should not be penalized. The arbiter interprets a low payment as a sign that the buyer is not ready to exploit the dataset—that is why their valuation is lower than the market price. w_i is computed as usual and applies to the next dataset. The buyer does not lose utility because they would not have won the dataset in the first place.

Gaming the market. To reduce risk, the ex-post algorithm should be activated only for returning buyers who have bought several datasets in the past and are expected to continue doing so, e.g., Chief Data Officers, and data hunters. Also, the ex-post algorithm may be activated only for specific datasets, for example, excluding high-value ones, hence bounding the potential revenue loss.

Risk-seeking buyers may bid for low-value datasets immediately after being assigned a wait-period, expecting their penalty will be consumed waiting for the low-period dataset and before they need to bid for a dataset they really want in the future. Extending the wait-period when they bid for *any* dataset may serve as a deterrent.

8.3 Balancing Ex-Post Balance

In ex-ante scenarios, the revenue balance (payments minus data value) is zero because data is only allocated when the payment is above the posting price. In ex-post scenarios, revenue may become negative after buyers pay below market price. This may deteriorate market performance, so the ex-post option should be deactivated for buyers whose balance is above a threshold.

To recover access to the ex-post option after losing it, buyers must pay the balance difference. The arbiter cannot tell buyers directly the magnitude of the difference because that is equivalent to leaking prices. Instead, on subsequent bids, arbiters pay the dataset price plus a fraction of their revenue balance. The arbiter does not release the magnitude of the fraction. Then, the revenue balance is *eventually* zero.

9 RELATED WORK

Although there is ample research in the theoretical design of data markets [4, 48], they do not explicitly target participants that strategize over time, and participants will strategize over time in data markets as the user study demonstrates. Dynamic mechanism design [18, 54] and repeated ad-auctions [22, 23] deal with strategic participants over time. Still, they do not target data market scenarios, e.g., ad space is rival, unlike data, that can be sold infinite times to many people. Query pricing work [15, 41] focuses on pricing queries over datasets for which a price has been set, but it does not offer a solution to finding the price of datasets in the first place. More importantly, there is an increasing number of work on data markets in the data management community for different kinds of models [44, 45]. Our work contributes to this growing area by bringing attention to the need for protecting data markets from strategic participants and proposing techniques to deal with strategic bueyrs.

Mechanism and Market Design. **Selling information.** There is a rich literature in auction and mechanism design for selling information [9], even in the presence of privacy constraints [31]. More recent work focuses on the unique characteristics of data as an asset [47] and on its characteristics. Our work focuses mainly on the kinds of online data marketplaces we see raising today [8, 21, 62, 63], and on providing techniques to protect against strategic behavior.

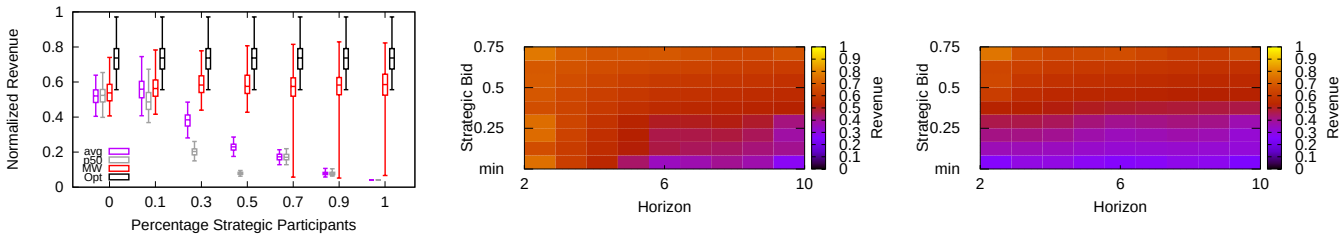

Figure 5: Left: Normalized revenue comparison of different update algorithms as PCT increases. Center: revenue as a function of horizon and strategic bid when PCT=0.5. Right: revenue as a function of horizon and strategic bid when PCT=0.9

Posting price. Posting price mechanisms for non-rivalry goods were used in the context of digital auctions [32, 33]. In this work, buyers bid exactly once and after the good is allocated no more bids are accepted. These mechanisms are designed for digital goods such as Netflix and Spotify, and not for markets of datasets, where streaming buyers strategize over time. Online learning has been used to choose posting prices [13, 40]. The emphasis of this work is to design high-performance online mechanisms. We focus on protecting markets against strategic buyers.

More recently, in the context of ad auctions, some work has made an explicit connection between posting price mechanisms and differential privacy [2] in order to protect the mechanism from overfitting to low bids. This paper focuses on data markets, where the good is freely replicable, unlike ad space, which is finite.

Dynamic mechanism design and revenue management. The fields of dynamic mechanism design [54] and revenue management [17, 18] consider strategic participants over time. The first often assumes buyers' value distribution is known. The second assumes goods are finite.

Uncertainty-Shield. Some work focuses on mechanism design in the presence of boundedly rational behavior [25]. Obvious strategy-proof mechanisms [10, 43] may ameliorate the boundedly-rational behavior we observed empirically and that may be associated with the complexity of the mechanisms. This work can complement the Uncertainty-Shield technique we propose here, which is otherwise simpler to incorporate in the algorithm.

Ex-post. There is work focuses on pricing information before the private value is known, whether by learning it on-the-fly [26], or by using tools from contract theory [51]. This research line complements the ex-post technique presented in this paper, which has been designed to be compatible with the pricing algorithm.

Data Markets. **Data Markets and Data Management Research.** The database community has produced work on data markets, with surveys of marketplaces [55, 59], vision papers such as DMMS [27] and Anylog [1]. The *query pricing* line of work [15, 16, 41] focuses on pricing a query over relational data while avoiding arbitrage opportunities. This line of work assumes relations have a set price, while our work complements it by focusing on the problem of finding that price in the first place.

A related problem the database community has tackled is this: given a function that associates a machine learning model's price with its accuracy, how to allocate the raised revenue to those who contributed data in the first place. Proposed solutions focus on adapting the Shapley value [36, 60] to this setting. Our contributions focus on pricing datasets, complementing this line of work.

Dealer [45] presents an end-to-end market design for machine learning models. Its main focus is on an arbitrage-free pricing algorithm and on revenue allocation. Our protection techniques can endow Dealer with protection against strategic buyer behavior.

A ML model fine-tuning market approach is presented in [44]. The paper proposes algorithmic contributions to navigate an exploration/exploitation tradeoff and obtain records that improve the target accuracy of a model the most given a target budget. If the techniques presented in [44] were implemented as part of a market, then the approach presented in our paper could be adapted to protect that market.

Theoretical Data Markets. The algorithmic marketplace [4] offers a model where the asset consists of ML models and where 1 buyer and 1 seller participate at a time. The emphasis of this work is on the modeling aspects and in a sampling method to speed up the computation of Shapley value, used for revenue allocation. More similar to our data market model is this work [48], which considers streaming buyers and sellers. Unlike them, we focus on studying how strategic buyers game the market and provide mechanisms to protect pricing algorithms from non-rational behavior. Finally, other work [22, 23] has studied the problem of dealing with strategic buyers that bid over time by providing truthful mechanisms, but their setting is ad auctions and not data markets.

10 CONCLUSION

The data management community has much to contribute to the practical design and implementation of data markets. This paper has brought attention to the need for protecting data markets from the strategic behavior that humans will introduce. The model presented in this paper is sufficiently general to represent many different types of data markets. The paper contributed 3 protection techniques and their integration into a pricing algorithm that we have evaluated extensively through simulations. The user study and simulations demonstrated the viability of these 3 techniques. All in all, the paper offered a first step towards practical data markets.

ACKNOWLEDGMENTS

We thank the reviewers for the valuable feedback, the user study participants, and the IRB for all the guidance offered.

REFERENCES

[1] Daniel J Abadi, Owen Arden, Faisal Nawab, and Moshe Shadmon. 2020. AnyLog: a Grand Unification of the Internet of Things.. In *CIDR*.

[2] Jacob D Abernethy, Rachel Cummings, Bhuvesh Kumar, Sam Taggart, and Jamie H Morgenstern. 2019. Learning Auctions with Robust Incentive Guarantees. In *Advances in Neural Information Processing Systems*. 11591–11601.

[3] Daron Acemoglu, Ali Makhdoumi, Azarakhsh Malekian, and Asuman Ozdaglar. 2019. *Too much data: Prices and inefficiencies in data markets*. Technical Report. National Bureau of Economic Research.

[4] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. 2019. A Marketplace for Data: An Algorithmic Solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation* (Phoenix, AZ, USA) *(EC '19)*. ACM, New York, NY, USA, 701–726. https://doi.org/10.1145/3328526.3329589

[5] Hirotugu Akaike. 1969. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics* 21, 1 (1969), 243–247.

[6] Sanjeev Arora, Elad Hazan, and Satyen Kale. 2012. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing* 8, 1 (2012), 121–164.

[7] Kenneth Joseph Arrow. 1972. Economic welfare and the allocation of resources for invention. In *Readings in industrial economics*. Springer, 219–236.

[8] awsmarket [n.d.]. AWS Data Exchange. https://aws.amazon.com/data-exchange/.

[9] Moshe Babaioff, Robert Kleinberg, and Renato Paes Leme. 2012. Optimal mechanisms for selling information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 92–109.

[10] Sophie Bade and Yannai A Gonczarowski. 2016. Gibbard-Satterthwaite success stories and obvious strategyproofness. *arXiv preprint arXiv:1610.04873* (2016).

[11] Ziv Bar-Yossef, Kirsten Hildrum, and Felix Wu. 2002. Incentive-compatible online auctions for digital goods. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 964–970.

[12] Avrim Blum. 1998. On-line algorithms in machine learning. In *Online algorithms*. Springer, 306–325.

[13] Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. 2004. Online learning in online auctions. *Theoretical Computer Science* 324, 2-3 (2004), 137–146.

[14] Michal Brzezinski. 2012. The Chen–Shapiro test for normality. *The Stata Journal* 12, 3 (2012), 368–374.

[15] Shuchi Chawla, Shaleen Deep, Paraschos Koutrisw, and Yifeng Teng. 2019. Revenue Maximization for Query Pricing. *Proc. VLDB Endow.* 13, 1 (Sept. 2019), 1–14. https://doi.org/10.14778/3357377.3357378

[16] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards Model-based Pricing for Machine Learning in a Data Marketplace. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. ACM, New York, NY, USA, 1535–1552. https://doi.org/10.1145/3299869.3300078

[17] Yiwei Chen and Vivek F Farias. 2018. Robust dynamic pricing with strategic customers. *Mathematics of Operations Research* 43, 4 (2018), 1119–1142.

[18] Yiwei Chen, Vivek F Farias, and Nikolaos Trichakis. 2019. On the efficacy of static prices for revenue management in the face of strategic customers. *Management Science* 65, 12 (2019), 5535–5555.

[19] Hoi Wai Jackie Cheng. 2020. Economic properties of data and the monopolistic tendencies of data economy: policies to limit an Orwellian possibility. (2020).

[20] Ralph B D'agostino, Albert Belanger, and Ralph B D'Agostino Jr. 1990. A suggestion for using powerful and informative tests of normality. *The American Statistician* 44, 4 (1990), 316–321.

[21] dawex [n.d.]. Dawex: Sell, buy and share data. https://www.dawex.com/en/.

[22] Alexey Drutsa. 2017. Horizon-independent optimal pricing in repeated auctions with truthful and strategic buyers. In *Proceedings of the 26th International Conference on World Wide Web*. 33–42.

[23] Alexey Drutsa. 2020. Optimal non-parametric learning in repeated contextual auctions with strategic buyer. ICML.

[24] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3–4 (Aug. 2014), 211–407. https://doi.org/10.1561/0400000042

[25] David Easley and Arpita Ghosh. 2015. Behavioral mechanism design: Optimal crowdsourcing contracts and prospect theory. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. 679–696.

[26] Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. 2018. Learning to bid without knowing your value. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 505–522.

[27] Raul Castro Fernandez, Pranav Subramaniam, and Michael J Franklin. 2020. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proceedings of the VLDB Endowment* 13, 11 (2020).

[28] Forbes. 2021. Data preparation is the most time-consuming and least enjoyable data science task according to survey. https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=5edde7236f63.

[29] GDPR. [n.d.]. General Data Protection Regulation. https://gdpr.eu.

[30] Arpita Ghosh, Katrina Ligett, Aaron Roth, and Grant Schoenebeck. 2014. Buying private data without verification. In *Proceedings of the fifteenth ACM conference on Economics and computation*. 931–948.

[31] Arpita Ghosh and Aaron Roth. 2011. Selling privacy at auction. In *Proceedings of the 12th ACM conference on Electronic commerce*. 199–208.

[32] Andrew V. Goldberg and Jason D. Hartline. 2001. Competitive Auctions for Multiple Digital Goods. In *Proceedings of the 9th Annual European Symposium on Algorithms (ESA '01)*. Springer-Verlag, Berlin, Heidelberg, 416–427.

[33] Andrew V. Goldberg and Jason D. Hartline. 2003. Envy-Free Auctions for Digital Goods. In *Proceedings of the 4th ACM Conference on Electronic Commerce* (San Diego, CA, USA) *(EC '03)*. Association for Computing Machinery, New York, NY, USA, 29–35. https://doi.org/10.1145/779928.779932

[34] Andrew V Goldberg, Jason D Hartline, Anna R Karlin, Michael Saks, and Andrew Wright. 2006. Competitive auctions. *Games and Economic Behavior* 55, 2 (2006), 242–269.

[35] Jason D Hartline. 2013. Mechanism design and approximation. *Book draft. October* 122 (2013).

[36] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, and Dawn Song. 2019. Efficient Task-specific Data Valuation for Nearest Neighbor Algorithms. *Proc. VLDB Endow.* 12, 11 (July 2019), 1610–1623. https://doi.org/10.14778/3342263.3342637

[37] Charles I Jones and Christopher Tonetti. 2019. *Nonrivalry and the Economics of Data*. Technical Report. National Bureau of Economic Research.

[38] John H Kagel, Ronald M Harstad, and Dan Levin. 1987. Information impact and allocation rules in auctions with affiliated private values: A laboratory study. *Econometrica: Journal of the Econometric Society* (1987), 1275–1304.

[39] Daniel Kahneman and Amos Tversky. 2013. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 99–127.

[40] Robert Kleinberg and Tom Leighton. 2003. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 594–605.

[41] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2015. Query-Based Data Pricing. *J. ACM* 62, 5, Article 43 (Nov. 2015), 44 pages. https://doi.org/10.1145/2770870

[42] Klaus Kultti. 1999. Equivalence of auctions and posted prices. *Games and Economic behavior* 27, 1 (1999), 106–113.

[43] Shengwu Li. 2017. Obviously strategy-proof mechanisms. *American Economic Review* 107, 11 (2017), 3257–87.

[44] Yifan Li, Xiaohui Yu, and Nick Koudas. 2021. Data Acquisition for Improving Machine Learning Models. *arXiv preprint arXiv:2105.14107* (2021).

[45] Jinfei Liu, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. 2021. Dealer: an end-to-end model marketplace with differential privacy. *Proceedings of the VLDB Endowment* 14, 6 (2021), 957–969.

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

[47] Sameer Mehta, Milind Dawande, Ganesh Janakiraman, and Vijay Mookerjee. 2019. How to sell a dataset? Pricing policies for data monetization. In *20th ACM Conference on Economics and Computation, EC 2019*. Association for Computing Machinery, Inc, 679.

[48] Dmitry Moor. 2019. Data Markets with Dynamic Arrival of Buyers and Sellers. In *Proceedings of the 14th Workshop on the Economics of Networks, Systems and Computation* (Phoenix, Arizona) *(NetEcon '19)*. Association for Computing Machinery, New York, NY, USA, Article 3, 6 pages. https://doi.org/10.1145/3338506.3340270

[49] Joan Morris, Peter Ree, and Pattie Maes. 2000. Sardine: Dynamic seller strategies in an auction marketplace. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*. 128–134.

[50] Roger B. Myerson. 1981. Optimal Auction Design. *Math. Oper. Res.* 6, 1 (Feb. 1981), 58–73. https://doi.org/10.1287/moor.6.1.58

[51] Parinaz Naghizadeh and Arunesh Sinha. 2019. Adversarial contract design for private data commercialization. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 681–699.

[52] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA.

[53] NIST. 2021. General Server Security. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-123.pdf.

[54] Alessandro Pavan, Ilya Segal, and Juuso Toikka. 2014. Dynamic mechanism design: A myersonian approach. *Econometrica* 82, 2 (2014), 601–653.

[55] Jian Pei. 2020. Data Pricing–From Economics to Data Science. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3553–3554.

[56] John W Pratt. 1959. Remarks on zeros and ties in the Wilcoxon signed rank procedures. *J. Amer. Statist. Assoc.* 54, 287 (1959), 655–667.

[57] Prolific. [n.d.]. Prolific. https://prolific.co.

[58] Alvin E Roth. 2008. What have we learned from market design? *The Economic Journal* 118, 527 (2008), 285–310.

[59] Fabian Schomm, Florian Stahl, and Gottfried Vossen. 2013. Marketplaces for data: an initial survey. *ACM SIGMOD Record* 42, 1 (2013), 15–26.

[60] Lloyd S. Shapley. 1952. A Value for n-Person Games. *Santa Monica, CA: RAND Corporation* (1952).
[61] Michael Stonebraker, Uundefinedur Çetintemel, and Stan Zdonik. 2005. The 8 Requirements of Real-Time Stream Processing. *SIGMOD Rec.* 34, 4 (Dec. 2005), 42–47. https://doi.org/10.1145/1107499.1107504
[62] worldquant [n.d.]. WorldQuant. https://data.worldquant.com.
[63] xignite [n.d.]. Xignite. https://aws.amazon.com/solutionspace/financial-services/solutions/xignite-market-data-cloud-platform/.