Demonstration of Dealer: An End-to-End Model Marketplace with Differential Privacy

Qiongqiong Lin, Jiayao Zhang, Jinfei Liu, Kui Ren[*]
Zhejiang University; Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province
{linqq,jiayaozhang,jinfeiliu,kuiren}@zju.edu.cn

Jian Lou
Emory University
jian.lou@emory.edu

Junxu Liu
Renmin University of China
junxu_liu@ruc.edu.cn

Li Xiong
Emory University
lxiong@emory.edu

Jian Pei
Simon Fraser University
jpei@cs.sfu.ca

Jimeng Sun
UIUC
jimeng@illinois.edu

ABSTRACT

Data-driven machine learning (ML) has witnessed great success across a variety of application domains. Since ML model training relies on a large amount of data, there is a growing demand for high-quality data to be collected for ML model training. Data markets can be employed to significantly facilitate data collection. In this work, we demonstrate Dealer, an en**D**-to-end mod**e**l m**a**rketp**l**ace with diff**e**rential p**r**ivacy. Dealer consists of three entities, data owners, the broker, and model buyers. Data owners receive compensation for their data usages allocated by the broker; The broker collects data from data owners, builds and sells models to model buyers; Model buyers buy their target models from the broker. We demonstrate the functionalities of the three participating entities and the abbreviated interactions between them. The demonstration allows the audience to understand and experience interactively the process of model trading. The audience can act as a data owner to control what and how the data would be compensated, can act as a broker to price machine learning models with maximum revenue, as well as can act as a model buyer to purchase target models that meet expectations.

PVLDB Reference Format:
Qiongqiong Lin, Jiayao Zhang, Jinfei Liu, Kui Ren, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. Demonstration of Dealer: An End-to-End Model Marketplace with Differential Privacy. PVLDB, 14(12): 2747-2750, 2021.
doi:10.14778/3476311.3476335

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/ZJU-DIVER/Dealer.

1 INTRODUCTION

Data-driven research, and more specifically machine learning, has witnessed substantial progress for multiple tasks and offers valuable potential to industries and businesses. High usability machine learning models depend on a large amount of high-quality training data, which makes it evident that data is a valuable resource. The commoditization of data has been approached in various ways. A data marketplace with model-based pricing provides a way for machine learning model instances to trade, which involves three parties, i.e., data owners, the broker, and model buyers.

- *Data owners.* Data owners receive compensation for allocating data usages to the broker. Meanwhile, they prefer to set limitations on privacy exposure when supplying their data to the broker. During the formulation of data owners' compensation functions, both the fair sharing of revenues allocated by the broker and requisites on privacy preservation should be accounted.
- *Broker.* The broker collects data from multiple data owners, designs and builds models, and then sells the models to multiple model buyers. Both data owners' compensation functions and model buyers' price functions should be taken into account by the broker when making market decisions. To maximize the revenue, the broker carefully prices a set of models with arbitrage-free guarantee and trains a set of models with maximum Shapley coverage, given a manufacturing budget to remain competitive.
- *Model buyers.* Model buyers always wish to buy cost-effective models that satisfy their demands. They report how much they are willing to pay for their target models. When formulating model buyers' price functions, both resistance on model noise and the relative utility of the model captured by the Shapley value of the data used to build the model should be accounted.

Recently, many efforts have been made to ensure the broker follows important market design principles in [1, 5, 6, 9]. Our prior work [7] proposes an en**D**-to-end mod**e**l m**a**rketp**l**ace with diff**e**rential p**r**ivacy (*Dealer*), which can simultaneously satisfy the needs of all three entities. In this demonstration, we apply those theoretical frameworks and build a prototype marketplace dedicated to machine learning models with differential privacy along the line of model-based pricing.

An illustration of *Dealer* is provided in Figure 1, which includes three entities (i.e., data owners, the broker, and model buyers) and

[*] The first two authors contributed equally; Jinfei Liu is the corresponding author.

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476335

their abbreviated interactions. From the perspective of data owners, *Dealer* proposes a unique compensation function for each data owner based on both privacy sensitivity and Shapley value. Privacy sensitivity of data owners is used to quantify data owners' risk tolerance associated with privacy exposure. Shapley value [8] is used to establish a fair revenue distribution mechanism. From the perspective of model buyers, *Dealer* proposes a unique price function based on both Shapley coverage sensitivity and noise sensitivity, which are uniformly measured by the level of differential privacy (DP) [4]. Shapley coverage sensitivity is used to formalize buyers' demands on Shapley coverage, which is derived from Shapley value. Noise sensitivity is used to formulate buyers' resistance on taking advantage of the marketplace by model noise. From the perspective of the broker, *Dealer* depicts the full marketplace dynamics through two important functions including: 1) *model pricing*, whose purpose is to maximize the revenue while following the market design principle of arbitrage-freeness; and 2) *model training*, whose purpose is to maximize Shapley coverage given a manufacturing budget for each model version to maintain competitive.

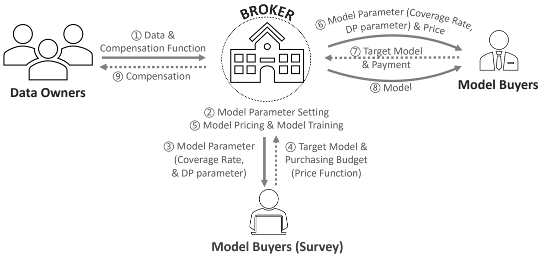

BROKER
Data Owners
① Data & Compensation Function
⑨ Compensation
⑥ Model Parameter (Coverage Rate, DP parameter) & Price
⑦ Target Model & Payment
⑧ Model
Model Buyers
② Model Parameter Setting
⑤ Model Pricing & Model Training
③ Model Parameter (Coverage Rate, & DP parameter)
④ Target Model & Purchasing Budget (Price Function)
Model Buyers (Survey)

Figure 1: Dealer framework.

2 DEALER: STRUCTURE AND PARTIES

In this section, we describe the structure of *Dealer* including three parties and their abbreviated interactions in the model marketplace.

2.1 Data Owners

Data owners share their data with the broker and expect compensation fairly based on the Shapley utility of the data contributed. We assume n data owners $\mathcal{D}_1, \ldots, \mathcal{D}_n$. Let b_i be a base price of user's data, ρ_i the user's privacy sensitivity, ϵ the differential privacy parameter. We use the following monotonic *compensation function* c_i for data owner D_i

$$c_i(\epsilon) = b_i \cdot (e^{\epsilon})^{\rho_i} = b_i \cdot e^{\rho_i \cdot \epsilon} \tag{1}$$

The base price b_i is proportional to the Shapley value of D_i with regard to its contribution to building models.

2.2 Model Buyers

Model buyers purchase models with budgets. Their offer prices are invariably linked to the coverage of the data used and noise added to the model.

Technically, Shapley value is utilized to measure the data coverage. Denote by $\mathcal{SV}(\mathcal{S})$ Shapley value of dataset $\mathcal{S}$. For a model M that is built using k datasets $D_{i_1}, \ldots, D_{i_k}$ from their data owners, the *Shapley coverage rate* of the model M is

$$\mathcal{CR}(M) = \mathcal{CR}(D_{i_1}, \ldots, D_{i_k}) = \frac{\mathcal{SV}(\{D_{i_1}, \ldots, D_{i_k}\})}{\mathcal{SV}(\{D_1, \ldots, D_n\})} \tag{2}$$

Putting the above two aspects together, for a buyer B_j $(1 \leq j \leq m)$, let V_j be the buyer's total budget to purchase a model, θ_j the expectation on Shapley coverage in Shapley value, δ_j the Shapley coverage sensitivity, η_j the expectation on model noise in differential privacy, and γ_j the noise sensitivity. Then, for a model M that is built using datasets $D_{i_1}, \ldots, D_{i_k}$ from data owners and satisfies ϵ-differential privacy, we use the following *price function* for model buyer B_j on model M

$$P(B_j, M) = V_j \cdot \frac{1}{1 + e^{-\delta_j (CR(M) - \theta_j)}} \cdot \frac{1}{1 + e^{-\gamma_j (\epsilon - \eta_j)}} \tag{3}$$

2.3 Broker

To prevent retail arbitrage, the broker investigates detailed information about buyers' purchase plans. The revenue maximization ($\mathcal{RM}$) problem is formulated as follows.

$$\arg \max_{\langle p(\epsilon_1), \ldots, p(\epsilon_l) \rangle} \sum_{k=1}^{l} \sum_{j=1}^{m'} p(\epsilon_k) \cdot \mathbb{I}(tm_j == M_k) \cdot \mathbb{I}(p(\epsilon_k) \leq v_j), \tag{4}$$

$$s.t.\ p(\epsilon_{k_1} + \epsilon_{k_2}) \leq p(\epsilon_{k_1}) + p(\epsilon_{k_2}),\ \epsilon_{k_1}, \epsilon_{k_2} > 0, \tag{5}$$

$$0 < p(\epsilon_{k_1}) \leq p(\epsilon_{k_2}),\ 0 < \epsilon_{k_1} \leq \epsilon_{k_2}, \tag{6}$$

where tm_j indicates the target model of model buyer B_j, v_j is the budget by the buyer for purchasing tm_j, $\mathbb{I}(tm_j == M_k)$ indicates whether B_j's target model is M_k, $\mathbb{I}(p(\epsilon_k) \leq v_j)$ indicates whether the price for model M_k (p_k or $p(\epsilon_k)$ from the DP parameter perspective) with DP parameter ϵ_k is less than or equal to the budget of B_j for purchasing M_k.

To control the substantial cost of building model instances, the broker attempts to train the best model for each model version to remain competitive with the limited manufacturing budget. Naturally, the optimal model version is the model that exploits a dataset under limited manufacturing budget with maximal Shapley coverage as follows.

$$\arg \max_{\mathcal{S} \subseteq \{D_1, \ldots, D_n\}} \sum_{i: D_i \in \mathcal{S}} \mathcal{SV}_i, \tag{7}$$

$$s.t. \sum_{i: D_i \in \mathcal{S}} c_i(\epsilon) \leq \mathcal{MB}. \tag{8}$$

where D_i indicates a data set of a data owner, $\mathcal{SV}_i$ is Shapley value of D_i, $\mathcal{S}$ indicates a subset of datasets, and $\mathcal{MB}$ indicates the manufacturing budget.

3 SYSTEM OVERVIEW

In this section, we introduce the architecture of *Dealer*, demonstrate the graphical user interfaces (GUIs) of the three entities, and explicitly explain the procedure of trading models. The system consists of two components: the front end and the back end.

3.1 Front End

The front end is implemented in JavaScript, which has the ability to send requests to the back end, get the response from the back end, and display the response to the audience. There are three modules for data owners, the broker, and model buyers, respectively.

Data owners' GUI. The GUI provides data owners with real-time interfaces to supply the broker with data and check out their compensation. Through the GUI, a data owner can submit her own data. She has the option to specify requisites on privacy preservation via entering privacy sensitivity that demonstrates the data owner's price elasticity of privacy. Due to the limited manufacturing budget of the broker, privacy sensitivity matters whether the data will be selected for model training. By inspecting Shapley value of the data, a data owner can gain direct insight into her data valuation among the participating data owners. When the data owners' data is used to build a model for sale, those who participate in model training will fairly share compensation based on Shapley value and privacy sensitivity.

Broker's GUI. The GUI supports the broker to arrange datasets from data owners and conduct transactions with model buyers. The broker first needs to select a dataset and choose a model type. Then, the broker conducts an elaborate market survey to collect the price functions of potential model buyers. *Dealer* helps the broker manage survey results and provides appropriate advice on model settings. Therefore, the broker enters model buyers' purchase budgets and privacy expectations through GUI. These market statistics will be sent to the back end later. After that, the front end will display a series of candidate model versions from the back-end response. The broker can choose whether to release the recommended models at the recommended price. Once the broker releases the models, the compensation prepaid by the broker will be sent to the data owners in real-time.

Model buyers' GUI. The released models with essential information (including DP parameter, Shapley coverage ratio, and price) will be rendered on the GUI of model buyers. The model buyers are allowed to browse various versions of models. Model buyers can choose a specific model version by submitting basic requirements (including Shapley coverage sensitivity and noise sensitivity). Their expectations and requirements will influence their purchase decisions. If a purchase decision is made, the model buyer needs to complete an online payment, after which the corresponding model instance can be downloaded freely through model buyers' GUI.

3.2 Back End

The back end is implemented in Python, which consists of four main modules: 1) compensation allocation, 2) model pricing, 3) model training, and 4) model suggestion.

Compensation allocation. The compensation allocation module is responsible for computing each data owner's actual compensation, as introduced in Section 2.1. Given the manufacturing budget, DP parameter, and privacy sensitivity, we fairly distribute the compensation for each participating data owner. To determine the base valuation of each data owner, we adopt Monte Carlo sampling to approximate Shapley value [2]. The larger the number of sampling permutations, the more accurate the computed Shapley value tends to be.

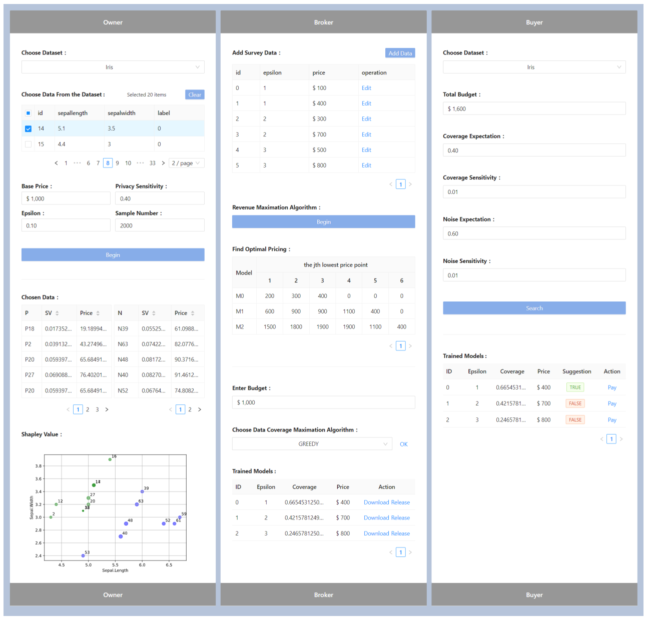
Model pricing. The model pricing module is implemented to quantify the price of the model with arbitrage-free constraint, and maximizes the revenue for the broker. Given a set of DP parameters and corresponding survey prices, the back end constructs a complete price space first, which converts an infinite price range into a set of discrete price points. Then the back end uses a dynamic programming algorithm to find optimal price solutions for model versions that can achieve maximum revenue. Please see [7] for details.

Model training. The main procedure of the model training module is to solve the Shapley coverage maximization problem, which is proved to be an NP-hardness problem. With both revenue maximization and Shapley coverage maximization, the broker will have a firm foothold in the market. Given the manufacturing budget, along with existing parameters (including each data owner's privacy sensitivity and Shapley value), we provide several optimal algorithms to (approximately) obtain datasets with maximum Shapley coverage. More details can be found in [7].

Model suggestion. Given a model buyer's total budget, Shapley coverage expectation, Shapley coverage sensitivity, noise expectation, and noise sensitivity, the model suggestion module induces the offer price of each model buyer for a model, as introduced in Section 2.2. Comparing model buyer's offer price with model's selling price, suggestions on model purchase for model buyers are further given.

4 SYSTEM DEMONSTRATION

In this demonstration, the audience can actively engage with various visual scenarios that showcase a fully functional implementation of *Dealer* as shown in Figure 2. The audience is allowed to personate data owners, the broker, or model buyers as she likes. The scenario of selling support-vector machine (SVM) models trained with iris dataset [3] is employed to demonstrate our system.



Owner

Choose Dataset : Iris

Choose Data From the Dataset : Selected 20 items Clear

■	id	sepallength	sepalwidth	label
☑	14	5.1	3.5	0
□	15	4.4	3	0

< 1 ··· 6 7 8 9 10 ··· 33 > 2 / page

Base Price : $ 1,000

Privacy Sensitivity : 0.40

Epsilon : 0.10

Sample Number : 2000

Begin

Chosen Data :

P	SV	Price	N	SV	Price
P18	0.017352...	19.18994...	N39	0.05525...	61.0988...
P2	0.039132...	43.27496...	N63	0.07422...	82.0776...
P20	0.059397...	65.68491...	N48	0.08172...	90.3716...
P27	0.069088...	76.40201...	N40	0.08270...	91.4612...
P20	0.059397...	65.68491...	N52	0.06764...	74.8082...

< 1 2 3 > < 1 2 >

Shapley Value :

Owner

Broker

Add Survey Data : Add Data

id	epsilon	price	operation
0	1	$ 100	Edit
1	1	$ 400	Edit
2	2	$ 300	Edit
3	2	$ 700	Edit
4	3	$ 500	Edit
5	3	$ 800	Edit

< 1 >

Revenue Maximation Algorithm : Begin

Find Optimal Pricing :

Model	the jth lowest price point					
	1	2	3	4	5	6
M0	200	300	400	0	0	0
M1	600	900	900	1100	400	0
M2	1500	1800	1900	1900	1100	400

< 1 >

Enter Budget : $ 1,000

Choose Data Coverage Maximation Algorithm : GREEDY OK

Trained Models :

ID	Epsilon	Coverage	Price	Action
0	1	0.6654531250...	$ 400	Download Release
1	2	0.4215781249...	$ 700	Download Release
2	3	0.2465781250...	$ 800	Download Release

< 1 >

Broker

Buyer

Choose Dataset : Iris

Total Budget : $ 1,600

Coverage Expectation : 0.40

Coverage Sensitivity : 0.01

Noise Expectation : 0.60

Noise Sensitivity : 0.01

Search

Trained Models :

ID	Epsilon	Coverage	Price	Suggestion	Action
0	1	0.6654531...	$ 400	TRUE	Pay
1	2	0.4215781...	$ 700	FALSE	Pay
2	3	0.2465781...	$ 800	FALSE	Pay

< 1 >

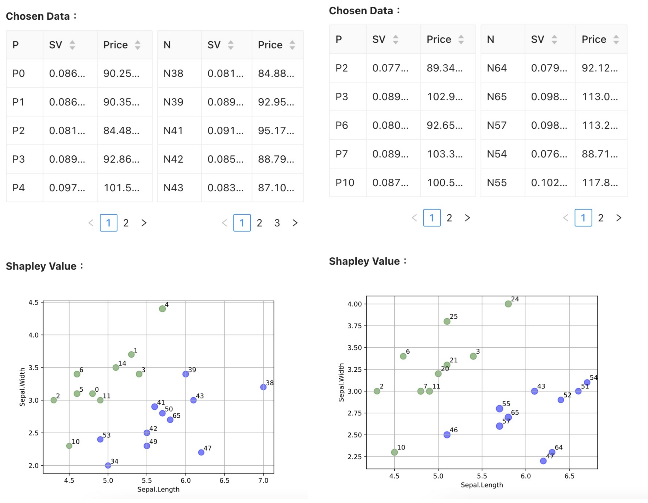
Buyer

Figure 2: GUIs of Dealer: the left panel visualizes compensation allocation of data owners; the middle panel visualizes model design of the broker; and the right panel visualizes model suggestions of model buyers.

4.1 Compensation of Data Owners

The audience simulates data owners' participation in the model marketplace. Each data tuple corresponds to a virtual data owner. The audience forms a coalition of data owners for model training via selecting some data tuples. Besides, the audience determines a parameter called data owners' privacy sensitivity. The compensation function of each data owner is sensitive to the preset privacy sensitivity and Shapley value of the data tuple.

We estimate Shapley value of the chosen data tuples first. To distinguish the utilities of different data tuples, we use a two-dimensional figure to show both data distribution and difference in Shapley value. For each data point in the figure, the larger its size is, the higher its Shapley value. Examples are shown in Figure 3. There are 20 data tuples that are randomly selected in the figure, including two groups with different labels. Then Shapley value and actual compensation are computed by the back-end functions. The same data tuple which corresponds to the same data owner has different Shapley value in different coalitions, while the data tuples near the hyperplane, e.g., support vectors, generally have higher Shapley value.



Chosen Data :

P	SV	Price
P0	0.086...	90.25...
P1	0.086...	90.35...
P2	0.081...	84.48...
P3	0.089...	92.86...
P4	0.097...	101.5...

N	SV	Price
N38	0.081...	84.88...
N39	0.089...	92.95...
N41	0.091...	95.17...
N42	0.085...	88.79...
N43	0.083...	87.10...

Shapley Value :

Chosen Data :

P	SV	Price
P2	0.077...	89.34...
P3	0.089...	102.9...
P6	0.080...	92.65...
P7	0.089...	103.3...
P10	0.087...	100.5...

N	SV	Price
N64	0.079...	92.12...
N65	0.098...	113.0...
N57	0.098...	113.2...
N54	0.076...	88.71...
N55	0.102...	117.8...

Shapley Value :

(a) Data coalition 1.

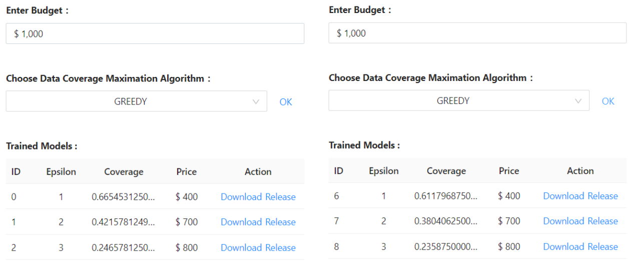
(b) Data coalition 2.

Figure 3: Shapley value of different data coalitions.

4.2 Model Design of the Broker

We showcase the procedure of model pricing and model training in our system that simulates a real model marketplace scenario. The audience collects market survey results, i.e., survey price points in our system. The back end solves the revenue maximization problem first; then figures out the Shapley coverage maximization problem with the constraint of the manufacturing budget. After the model design results are rendered in the front end, the audience can determine whether to release the displayed models.

Naturally, a data owner's privacy sensitivity affects her compensation function, further affects whether her data will be selected into the dataset for model training. Samples are shown in Figure 4. Under the same manufacturing budget, different privacy sensitivity of data owners will lead to different datasets that achieve the optimal Shapley coverage, which is reflected by the coverage property of the trained models in Figure 4.



Enter Budget :

$ 1,000

Choose Data Coverage Maximation Algorithm :

GREEDY

Trained Models :

ID	Epsilon	Coverage	Price	Action
0	1	0.6654531250...	$ 400	Download Release
1	2	0.4215781249...	$ 700	Download Release
2	3	0.2465781250...	$ 800	Download Release

Enter Budget :

$ 1,000

Choose Data Coverage Maximation Algorithm :

GREEDY

Trained Models :

ID	Epsilon	Coverage	Price	Action
6	1	0.6117968750...	$ 400	Download Release
7	2	0.3804062500...	$ 700	Download Release
8	3	0.2358750000...	$ 800	Download Release

(a) Privacy sensitivity = 0.1

(b) Privacy sensitivity = 0.2

Figure 4: Model design with different privacy sensitivity.

4.3 Model Suggestion of Model Buyers

Similar to the previous compensation of data owners, we also allow the audience to give an expected price and expectations of other model properties towards model instances for version screening. A real-time query system is supported by the back end, which filters out the trained models that satisfy the requirements. The query results and tips will be listed in a table. In Column "Suggestion" of Figure 2, green "TRUE" indicates that the corresponding model meets the preset conditions, while red "FALSE" indicates that it is not recommended to buy. The audience can further click the suggestion to complete an online payment.

ACKNOWLEDGMENTS

This work was supported in part by the Fundamental Research Funds for the Central Universities, the NSF grants (CNS-1952192, CNS-2027783, SCH-2014438, PPoSS 2028839, and IIS-1838042), AFOSR under DDDAS program FA9550-12-1-0240, the NSFC grants (61941121 and 91646203), the NSERC Discovery Grant program, and the NIH grants (R01 1R01NS107291-01 and R56HL138415).

REFERENCES

[1] 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*. ACM, 701–726.

[2] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & OR* 36, 5 (2009), 1726–1730. https://doi.org/10.1016/j.cor.2008.04.004

[3] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

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

[5] Raul Castro Fernandez, Pranav Subramaniam, and Michael J. Franklin. 2020. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proc. VLDB Endow.* 13, 11 (2020), 1933–1947. http://www.vldb.org/pvldb/vol13/p1933-fernandez.pdf

[6] 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. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1610–1623.

[7] Jinfei Liu, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. 2021. Dealer: An End-to-End Model Marketplace with Differential Privacy. *Proc. VLDB Endow.* 14, 6 (2021), 957–969.

[8] Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[9] Tianshu Song, Yongxin Tong, and Shuyue Wei. 2019. Profit Allocation for Federated Learning. In *2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, December 9-12, 2019*. IEEE, 2577–2586. https://doi.org/10.1109/BigData47090.2019.9006327