Game of Privacy: Towards Better Federated Platform Collaboration under Privacy Restriction

Chuhan Wu\textsuperscript{1}, Fangzhao Wu\textsuperscript{2}, Tao Qi\textsuperscript{1}, Yanlin Wang\textsuperscript{2}, Yuqing Yang\textsuperscript{2}, Yongfeng Huang\textsuperscript{1}, Xing Xie\textsuperscript{2}

\textsuperscript{1}Department of Electronic Engineering, Tsinghua University, Beijing 100084
\textsuperscript{2}Microsoft Research Asia, Beijing 100080, China

\{wuchuhan15,wufangzhao,taoqi.qt\}@gmail.com,yfhuang@tsinghua.edu.cn,{yanlwang,yuqing.yang,xingx}@microsoft.com

ABSTRACT

Vertical federated learning (VFL) aims to train models from cross-silo data with different feature spaces stored on different platforms. Existing VFL methods usually assume all data on each platform can be used for model training. However, due to the intrinsic privacy risks of federated learning, the total amount of involved data may be constrained. In addition, existing VFL studies usually assume only one platform has task labels and can benefit from the collaboration, making it difficult to attract other platforms to join in the collaborative learning. In this paper, we study the platform collaboration problem in VFL under privacy constraints. We propose to incentivize platforms through a reciprocal collaboration, where all platforms can exploit multi-platform information in the VFL framework to benefit their own tasks. With limited privacy budgets, each platform needs to wisely allocate its data quotas for collaboration with other platforms. Thereby, they naturally form a multi-party game. There are two core problems in this game, i.e., how to appraise other platforms’ data value to compute game rewards and how to optimize policies to solve the game. To evaluate the contributions of other platforms’ data, each platform offers a small amount of “deposit” data to participate in the VFL. We propose a performance estimation method to predict the expected model performance when involving different amount combinations of inter-platform data. To solve the game, we propose a platform negotiation method that simulates the bargaining among platforms and locally optimizes their policies via gradient descent. Extensive experiments on two real-world datasets show that our approach can effectively facilitate the collaborative exploitation of multi-platform data in VFL under privacy restrictions.

KEYWORDS

Federated learning, Privacy, Game, Collaboration

ACM Reference Format:

Chuhan Wu, Fangzhao Wu, Tao Qi, Yanlin Wang, Yuqing Yang, Yongfeng Huang, Xing Xie. 2022. Game of Privacy: Towards Better Federated Platform Collaboration under Privacy Restriction. In Proceedings of the 28th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’22), August 14–18, 2022, Washington D.C., ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnn.nnnnnn

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

KDD ’22, August 14–18, 2022, Washington D.C.
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnn.nnnnnn

1 INTRODUCTION

Federated learning (FL) \cite{20} offers the potential to learn intelligent models from decentralized data under privacy protection \cite{3}. A typical application of federated learning is leveraging multi-platform data for collaborative model training, where the samples are aligned across platforms while different feature fields of the same samples are kept by different platforms \cite{10}. This paradigm is usually named vertical federated learning (VFL) \cite{39}. Existing researches on VFL usually assume that all data on each platform can participate in the VFL coordinated by a target platform \cite{7,35}. However, in real-world scenarios, the amount of data to be involved in VFL may be limited due to privacy concerns and some strict regulations such as GDPR \cite{28}. The privacy budgets\textsuperscript{1} of data owners may also constrain their participation in federated learning \cite{39}. Moreover, existing VFL methods only benefit the platforms with sample labels of target task \cite{7}. They cannot provide sufficient incentives for other participants that provide data and computing resources \cite{8}. Thus, there is a huge gap between existing research works and real-world applications of vertical federated learning.

An intuitive way to incent participant platforms is to allocate profits according to their contributions \cite{27}. Unfortunately, there is no sufficient guarantee that the leader platform is honest and the profit allocation is fair to each platform \cite{42}. Instead of currency payment, we propose to incent the VFL participants via reciprocal collaborations \cite{6}, where each platform can benefit its target task from the complementary information encoded in cross-platform data \cite{26}, as shown in Fig. 1. Intuitively, a platform can usually better attract another platform’s collaboration if it offers more data for

\textsuperscript{1}We define a platform’s privacy budget as its maximum accumulated amount of data involved in VFL.
federated learning. However, the privacy budgets of platforms are very limited. Since collaborating with different platforms may lead to very different returns [31], platforms need to smartly allocate their data quotas for collaboration with other platforms. In this way, the reciprocal collaborations among platforms can naturally form a multi-party game, where each platform hopes to obtain a higher model performance gain with possibly less offered data by carefully designing its data quota allocation plan through the game. The incentives of platforms are obtained by the collaborations and competitions in this game, rather than passively allocated by a leader platform. The game among platforms can push them to optimize their collaboration policies, which can improve their ability in exploiting multi-platform data via VFL.

In this paper, we study the problem of platform collaboration in vertical federated learning under privacy restrictions. We propose to model the incentive problem in VFL as the aforementioned reciprocal collaboration problem, which can be regarded as a multi-party game with privacy budget restrictions. We propose a FedGame\(^2\) approach to solve this game. First, to compute platforms’ rewards in the game, we propose a performance estimation method to predict each platform’s model performance under different collaboration policies. More specifically, each platform first offers a small amount of “deposit” data to participate in the reciprocal vertical federated learning.\(^3\) Each platform leverages different amounts of inter-platform deposit data to train and evaluate multiple models, and uses multi-variable regression to estimate the real performance gains when involving different amounts of data from other platforms. Given the performance estimation results, we further propose a platform negotiation method to solve the game by simulating the bargaining behaviors in real-world commercial collaborations that negotiate the “price” of provided resources. Each platform first derives a reward from the expected performance gains and the amounts of data involved in the FL for other platforms, and then performs gradient descent to optimize the reward. The game among different platforms will converge after multiple rounds of negotiation, and platforms’ collaboration policies can be finalized for subsequent collaborative learning. Extensive experiments on two datasets in different scenarios show that FedGame can generate effective platform collaboration policies for VFL to empower the privacy-preserving exploitation of multi-platform data.

The contributions of this paper are listed as follows:

- To our best knowledge, this is the first work that studies the competitive collaboration among platforms in vertical federated learning under realistic privacy restrictions.
- We propose a performance estimation method to help platforms predict the performance gains under different amounts of inter-platform data involved in federated learning.
- We propose a platform negotiation mechanism to solve the multi-party game by simulating the bargaining behaviors in real-world platform collaborations.
- We conduct experiments on two datasets for different scenarios to verify the effectiveness of our approach in improving platforms’ collaborations in vertical federated learning.

\(^2\)Source code is available at https://github.com/wuch15/FedGame.

\(^3\)The intermediate results encoded by the local models and their gradients are exchanged, rather than the raw data.

2 RELATED WORK

Federated learning is a privacy-aware machine learning paradigm that can leverage highly decentralized private data to learn intelligent models collaboratively [14]. Instead of directly collecting and exchanging private user data for centralized model learning, in federated learning, only the intermediate model updates and variables are exchanged among a server and a number of clients, and thereby user privacy can be protected to a certain extent [20]. In the standard federated learning, different samples are kept by different clients and their input feature spaces are usually aligned [15]. This scenario is known as horizontal federated learning [39], which has been used in various scenarios such as medical data processing [12, 25], keyboard prediction [9], and personalized recommendation [24, 32, 38].

A major variant of the above paradigm is vertical federated learning (VFL) [44], which is employed by many methods to exploit multi-platform data for collaborative model training [1, 5, 16, 37]. For example, Hardy et al. [10] proposed a privacy-preserving logistic regression method that introduces a third-party server to coordinate the model learning on multi-platform data. Yang et al. [40] further proposed an improved version of vertical federated logistic regression that is free from the additional coordinator server. Wu et al. [33] proposed a federated ad CTR prediction approach that can exploit user behavior data on multiple platforms in a privacy-preserving way to empower both model training and inference. Zhang et al. [45] proposed an efficient bi-level VFL method that supports asynchronous model updates on all platforms. In these VFL methods, the samples on different platforms are usually aligned by privacy-preserving entity resolution [23], while the feature spaces of the same samples are decentralized on different platforms. Different from horizontal federated learning methods that learn fully or partially shared models to serve all clients, existing VFL methods mainly benefit the coordinator platform that holds labels in target tasks [7]. Since other participant platforms need to offer their data and computing resources that lead to additional costs, appropriate incentives mechanisms are necessary in VFL to attract platforms to join the collaborative learning [42, 46].

Unfortunately, most studies focus on incentives in horizontal federated learning [18, 22, 27, 36, 41, 43], while the incentive problem in VFL is rarely studied. Only a few works explore a relevant problem, i.e., the contribution evaluation in VFL [4, 8, 29, 30]. For example, Wang et al. [30] proposed to use Shapley value to measure the contribution of each party to VFL. Han et al. [8] proposed an information theory-based metric that modifies the Shapley value computation to assess data values of different platforms from a game-theoretic perspective. Fan et al. [4] proposed a vertical federated Shapley value measurement that can efficiently measure platform contribution under various VFL algorithms. These methods usually assume that all data on each platform can be involved in collaborative learning. In fact, due to the restriction of certain data protection regulations, the total amounts of data that can be exploited by federated learning may be strictly limited [28]. In addition, the intrinsic privacy risks of VFL [11, 17] may also narrow its use given the limited privacy budgets of data owners. Thus, the collaborations among platforms under privacy restrictions can be quite competitive. To design profitable collaboration policies that can obtain higher model performance gains, platforms need to
accurately estimate the value of incorporating different amounts of inter-platform data, which existing methods cannot achieve.

Different from existing works, we model the incentive problem in VFL as a reciprocal collaboration, where all platforms aim to maximally improve the model performance in their own tasks through VFL collaborations. Due to the restriction of platforms’ privacy budgets, we further form the reciprocal collaboration as a multi-party game. To measure other platforms’ contributions to one platform in the game, we propose a regression-based performance estimation method to measure the expected performance gains when incorporating different amounts of cross-platform data, which is distinct from existing Shapley value-based party contribution measurements that are agnostic to data volume. In addition, we propose a platform negotiation approach to solve the game and generate beneficial policies to facilitate the exploitation of multi-platform data. Our approach can facilitate the application of VFL in real-world commercial collaborations among multiple platforms.

3 METHODOLOGY

In this section, we introduce the details of our game-based VFL approach named FedGame, which can achieve competitive collaborations among platforms under privacy restrictions to effectively empower their target tasks with limited cross-platform data. We first introduce the problem formulation and basic assumptions of this work, and then present the details of our method.

3.1 Problem Formulation and Assumptions

In this work, we assume that there are N platforms that keep different types of private data, and sample IDs have been aligned across different platforms. Each platform has a target task, such as news recommendation and Ads CTR prediction, and the sample labels of this task are only kept on this platform. We define the privacy budget of a platform as the maximum accumulated amount of data it can provide to join the VFL coordinated by other platforms, which is denoted as $c_i$ for the $i$-th platform. Due to privacy restrictions, the privacy budget of each platform is limited. To collaborate with other platforms effectively under privacy restrictions, each platform has a policy to allocate different data quotas for the VFL on different platforms. We denote the amount of data offered by the $i$-th platform to join the VFL on the $j$-th platform as $c_{i,j}$, which satisfies the constraint $\sum_{j=1}^{N} c_{i,j} \leq c_i$ ($c_{i,i} = 0$). The policy of the $i$-th platform is defined as the data quota allocations for all other platforms, which is denoted as $\pi_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,N})$. The platform reward $r_i$ is defined as the expected model performance gain in the target task on the $i$-th platform, which depends on the amount of inter-platform data provided by other platforms. The goal of each platform is to maximize its reward under given privacy budget by optimizing its collaboration policy. In our scenario, the collaborative negotiations among platforms naturally form a multi-party game [34]. The different platforms are the agents in the game, and their permitted action is adjusting the amounts of data participated in other platforms’ VFL. The platforms can obtain their collaboration policies by solving this game, and they will conduct the final VFL on the allocated data based on platforms’ policies.

3.2 Overall Framework of FedGame

The overall framework of our FedGame approach is shown in Fig. 2. It simulates the negotiation behaviors among different platforms in the game. In each round of negotiation, different platforms first exchange their collaboration policies. Based on the shared policies, each platform evaluates the value of its own policy by computing the expected reward it can obtain. Then these platforms locally update their policy to optimize their expected rewards. If the optimization needs to continue, all platforms return to the collaboration policy exchanging stage and repeat the above steps. If the policy optimization of all platforms has converged, the platforms reach a consensus about the collaboration plans in VFL, and finally learn their models according to the finalized plans. We can see that there are two core problems in this negotiation framework:

- How to derive the reward of each platform to evaluate its current policy?
- How to optimize the platforms’ policies to solve the game?

To address these problems, we propose an accurate performance estimation method and an effective platform negotiation mechanism, respectively. We discuss the details in the following sections.

3.3 Performance Estimation

In existing methods, the contributions of different platforms are usually evaluated by Shapley value-based measurements [29, 30].
However, Shapley values do not necessarily indicate the real performance contribution [19, 21] and they are agnostic to the data amounts. Thus, they cannot effectively evaluate the real data value with changeable data sizes. To solve this problem, we propose a simple yet effective performance estimation method. Its framework is shown in Fig. 3. The core idea of our method is to learn a performance regression function, which models the relations between the amount of involved data on different platforms and the expected model performance. Its details are introduced as follows.

In our approach, each platform first offers a small amount (e.g., 5%) of “deposit” data randomly sampled from the local database to participate in the VFL. All platforms use their own local data and the useful information of the deposit data from other platforms to train models for their target tasks, and evaluate the model performance based on their local labeled test data. To accurately model the relation between model performance and the amount of involved data, each platform independently learns and evaluates models multiple times by randomly sampling different fractions of deposit data from different platforms to participate in the federated learning. Assume that there are $K$ experiments on the platform $i$, each of which has a data amount combination $(c_{1,i}, c_{2,i}, ..., c_{N,i})$ and a performance observation $r^k_i$: For this platform, it locally solves a regression problem $\hat{r}_i = f(c_{1,i}, c_{2,i}, ..., c_{N,i})$ based on the $K$ observations, where the independent variables are the amounts of involved data on different platforms, $\hat{r}_i$ is the expected model performance, and $f(\cdot)$ is a regression function. Since the model performance is usually better when more data is incorporated, the function $f$ should be monotonic. Thus, multi-variable linear regression may be a good option due to its simplicity and low requirements of the experiment times $K$ (it only needs to be larger than $N$). By inputting the $K$ performance observations into the Least Squares Regression method, we can obtain the relations between the observed model performance on each platform and the amounts of involved inter-platform data. We regress the model performance for all platforms to obtain the following regression functions:

$$\hat{r}_1 = b_1 + w_{1,1}c_{1,1} + ... + w_{N,1}c_{N,1},$$

$$\hat{r}_2 = b_2 + w_{1,2}c_{1,2} + ... + w_{N,2}c_{N,2},$$

$$...$$

$$\hat{r}_N = b_N + w_{1,N}c_{1,N} + ... + w_{N,N}c_{N,N},$$

where $\hat{r}_i$ is the estimated model performance on the $i$-th platform, $b_i$ is its basic model performance without inter-platform data, $w_{j,i}$ indicates its relative model performance improvement brought by a unit amount of data on the $j$-th platform. Using the regression results, we can obtain the estimated model performance gain on each platform given any amount combination of cross-platform data used in model training.

### 3.4 Game-based Platform Negotiation

Based on the performance estimation method introduced above, the platforms can evaluate their rewards given other platforms’ collaboration policies. However, the reward formulation in Eq. (1) cannot be directly optimized, since a platform cannot control the amount of data offered by other platforms (e.g., platform 1 cannot determine $c_{2,1}$). Although the amounts of data mutually provided by two platforms are correlated, their latent relations are unknown. Motivated by the concept of output/input ratio [2] in many real-world scenarios, we propose a reward reformulation method to convert platforms’ rewards into differentiable forms as follows:

$$r_1 = b_1 + \frac{(w_{2,1}c_{2,1})^Y}{c_{1,2} + \epsilon} + ... + \frac{(w_{N,1}c_{N,1})^Y}{c_{1,N} + \epsilon},$$

$$r_2 = b_2 + \frac{(w_{2,1}c_{2,1})^Y}{c_{2,1} + \epsilon} + ... + \frac{(w_{N,2}c_{N,2})^Y}{c_{2,N} + \epsilon},$$

$$...$$

$$r_N = b_N + \frac{(w_{1,N}c_{1,N})^Y}{c_{N,1} + \epsilon} + ... + \frac{(w_{N,N}c_{N,N})^Y}{c_{N,N} + \epsilon},$$

where $\gamma$ is a hyperparameter that indicates the preference for outputs against inputs, and $\epsilon$ is a small positive constant (e.g., 1e-8) to avoid division by zero. In these formulas, the reward of a platform is higher if it can use less data to “barter” more data on other platforms. In addition, platforms tend to allocate uniform data quotas (i.e., the same quotas for all other platforms) if $\gamma$ is smaller, while prefer platforms that have greater data contributions to their tasks when a larger value of $\gamma$ is used.

Although $\gamma$ can be empirically tuned, we can have an initial prior estimation of its suitable value to reduce the hyperparameter search effort in practical use. We can first assume that $c_{i,j}$ is approximately proportional to $c_{j,i}$, which means that contributing more to another platform can obtain more benefits from it. Since in Eq. (1), the original reward is also proportional to the data amounts, the value of $\gamma$ should be 2 to achieve the same degree of data amount variables as Eq. (1). In fact, due to the constraint of the total privacy budget, the same increase of $c_{i,j}$ may yield a smaller increase in $c_{j,i}$, which means that the correlation between a large $c_{i,j}$ and its associated $c_{j,i}$ may be ultra-linear. Therefore, the expected value of $\gamma$ is somewhat

---

1. The deposit data also counts in the privacy budget.
2. We assume that all local data on each platform is used for learning its own model.
3. We do not use more complicated regression methods such as high-order polynomial regression and exponential regression because they have poor accuracy when generalizing to the full data.
larger than 2 to achieve the same degree. In our experiments the optimal \( y \) value is about 2.5, which well matches the above analysis.

Finally, we discuss how to solve the game. In the policy exchange stage in Fig. 2, each platform can only access its concerned collaboration policies (e.g., \( c_{i,j} \)), but cannot see other irrelevant collaboration policies (e.g., \( c_{j,k} \)). Thus, the negotiation is in fact an incomplete information game. For simplicity, each platform can optimize its own policy based on the information it can access without guessing other platforms’ unknown behaviors. More specifically, each platform uses gradient descent to locally update its policy that can optimize its reward. The policy \( \pi_i \) of the \( i \)-th platform is updated as follows:

\[
\pi'_i = \pi_i - \eta \left( \frac{\partial r_i}{c_{i,1}}, \frac{\partial r_i}{c_{i,2}}, ..., \frac{\partial r_i}{c_{i,N}} \right),
\]

where \( \pi'_i \) is the updated policy, and \( \eta \) is a game learning rate. To ensure the updated policy satisfies the privacy constraint, each of its element \( c_{i,j} \) is further normalized as follows:

\[
c_{i,j} = \begin{cases} 
\max (0, c_{i,j}) & \text{if } \sum_j c_{i,j} \leq c_i, \\
\frac{c_i \max (0, c_{i,j})}{\sum_j c_{i,j}} & \text{if } \sum_j c_{i,j} > c_i.
\end{cases}
\]

It means that all elements are non-negative and their summation does not exceed the budget \( c_i \). After multiple rounds of negotiation in Fig. 2, we regard that the game is converged if the policy updates on all platforms satisfy \( ||\pi'_i - \pi_i|| < \mu \), where \( \mu \) is a small value indicating the stop criteria. When the platforms’ collaboration policies have been determined, they offer corresponding quotas of data to participate in the VFL launched by other platforms as well as train their own models based on the cross-platform data.

4 EXPERIMENTS

To validate the effectiveness of FedGame in solving the platform collaboration problem in VFL under privacy restrictions, we design our experiments to address the following research questions:

- RQ1: Does the game-based platform collaboration method outperform other baseline strategies?
- RQ2: Does the performance estimation method accurately predict model performance under different data quota combinations?
- RQ3: What are the collaboration policies obtained by the game solution?
- RQ4: How does the output/input preference \( y \) influence platform policies and model performance?
- RQ5: How does the amount of deposit data influence the collaboration effectiveness?

In the following sections, we first introduce the datasets and experimental settings, and then present the experimental results to answer each of the research questions above.

4.1 Datasets and Experimental Settings

We conduct extensive experiments on two datasets. The first one is Adult [13], which is a widely used tabular dataset. There are 15 feature fields for each sample. We divide them into three groups according to their orders in the dataset to simulate the scenario that each platform keeps 5 feature fields of a sample. On each simulated platform one type of feature is used as the prediction target and the rest are used as inputs (see supplements for detailed feature lists). The target tasks on the three platforms are education prediction (16-way classification, platform 1), gender prediction (binary classification, platform 2), and income prediction (binary classification, platform 3). The second dataset is the Ad CTR prediction dataset used in [33]. We denote this dataset as CTR. Following the settings in [33], we assume that different types of user behaviors in this dataset, including historical clicked Ads, search queries, and browsed webpages, are kept by three different platforms (clicked Ads on platform 1, search queries on platform 2, and browsed webpages on platform 3). The target task on each platform is to predict whether a candidate Ad will be clicked by a target user based on his/her behaviors. The statistics of the two datasets are listed in Table 1.

| Table 1: Statistics of the Adult and CTR datasets. |
|---------------------------------|
| Adult                          |
| #training samples              | 32,561 |
| #test samples                  | 16,281 |
| CTR                            |
| #users                         | 100,000 |
| avg. #queries per user         | 50.69  |
| #ads                           | 8,105  |
| avg. #webpages per user        | 210.09 |
| #ad clicks                     | 345,264|
| avg. #words per ad title       | 3.73   |

In our experiments, without loss of generality, we assume the budget of each platform is the amount of the training samples and we use normalized budgets for all platforms, which means that \( c_1 = c_2 = \cdots = c_N = 1 \) and the data amount variables \( (c_{i,j}) \) indicate the ratios of budget consumption. The number of independent experiments for performance estimation is 5. We use logistic regression on Adult and FedCTR [33] on CTR as the backbone models. For the logistic regression algorithm, we use the default settings in Sklearn. For the FedCTR method we follow its original settings. In our platform negotiation process, the learning rate \( \eta \) is 0.01. The value of \( y \) is 2.5. The small value \( \epsilon \) is 1e-8. We use accuracy as the metric on Adult and use AUC on CTR. We repeat each experiment 5 times and report the average results with standard deviations.

4.2 Performance Evaluation

To answer RQ1, we verify the effectiveness of FedGame in helping platforms’ collaborations. We compare the model performance on each platform as well as the platform average performance using different collaboration strategies, including: (1) Local data only, where all platforms learn their models solely based on local data; (2) Uniform policy, where each platform uniformly assigns data quotas for other platforms; (3) Greedy policy, where each platform chooses to contribute all its data budget to the platform with the highest value to it; (4) Shapley value-based policy, which uses deposit data to compute Shapley values and allocate data quotas accordingly. (5) FedGame, our game-based platform negotiation method. The

results are shown in Fig. 4. We have several interesting observations. First, compared with the local data only method, all other methods can improve the average model performance of platforms. This shows that leveraging multi-platform data can provide rich complementary information to support model training and serving. In addition, we find that the uniform policy prefers platforms with low data values (e.g., platform 1 on CTR since its model performance based on local data in the same task is the lowest), while is suboptimal for other platforms. Thus, its average performance is suboptimal for other platforms. Thus, its average performance is not optimal. By contrast, we find that the greedy policy makes some platforms excluded from the collaboration (platform 1 does not obtain any data, while platforms 2 and 3 contribute all their data to each other). Thus, the average performance is also suboptimal. Besides, the Shapley value-based policy does not perform very well. This is because Shapley values do not indicate the real performance gains, and allocating data quotas based on Shapley values is not very effective. Finally, our FedGame approach achieves the best average performance (t-test \( p < 0.01 \)), and it can even achieve better results than using greedy policy on most platforms. This shows that using appropriate policies can improve the collaborations between platforms to better exploit multi-platform data.

4.3 Results of Performance Estimation

To answer RQ2, we first show the regression coefficients computed by different platforms in Fig. 5. We find although data on different platforms have diverse values for different target tasks, incorporating the data from each platform in both datasets has a positive impact. Thus, all platforms in both scenarios are qualified to participate in the game. Since the performance regression is conducted based on the small amount of deposit data, we need to verify its generality to larger amounts of data. Thus, we compare the predicted and real model performance under different amount combinations of inter-platform data, as shown in Fig. 6. We can see that the predicted performance well matches the real

![Figure 4: Performance evaluation results on Adult. Error bars indicate standard deviations.](image)

![Figure 5: Performance regression coefficients on the two datasets. The value at the \( i \)-th row and \( j \)-th column is \( w_{j,i} \).](image)
performance on all platforms under most amount combinations of involved data. Thus, if the finally used data amount does not have an extremely huge gap (e.g., 1000 times) with the amount of deposit data, the model performance can be accurately estimated, and the computed rewards can be used as direct indications of potential model performance gains.

4.4 Results of Platform Collaboration Policy

Based on the performance estimation results, we solve the game among different platforms. To respond to RQ3, we show the final collaboration policies on the two datasets in Fig. 7. We have several interesting observations from the policies. First, we find that all platforms exhaust their privacy budget. This shows that platforms would like to offer more data to obtain more model performance gains if the privacy budget permits. In addition, we observe that all platforms tend to collaborate more with the parties that keep more valuable data to their target tasks. For example, on the Adult dataset, the first platform chooses to offer most of its data to the third platform, since the third platform’s data can effectively improve its model performance. Meanwhile, different from the greedy policy, the third platform still provides a part of its data for the first platform in return, which shows that the platforms have a valid collaboration. Similar phenomena also exist on the CTR dataset. These results imply that in our method, platforms do not fall into the prisoner’s dilemma where all platforms do not provide any data or some platforms are left out from the collaboration. Our proposed game-based platform negotiation method can facilitate the effective collaborations among different platforms to exploit multi-platform data under privacy restrictions.

4.5 Hyperparameter Analysis

To answer RQ4, we illustrate the model accuracy and policy of each platform under different values of \( \gamma \), as shown in Fig. 8. We find that platforms’ policies are more uniform when \( \gamma \) is smaller, while are more aggressive when \( \gamma \) is larger. Note that the policy does not change when \( \gamma \) is larger than 3 on Adult and 3.5 on CTR, where both the first and second platforms provide all their data for

![Figure 6: Real and predicted model performance.](image)

![Figure 7: The collaboration policies of different platforms. The value at the \( i \)-th row and \( j \)-th column is \( c_{i,j} \).](image)
Figure 8: The influence of $\gamma$ on platforms’ policies and final model performance. The dashed lines represent the ratios of data offered to other platforms. The best average performance on both datasets is achieved when $\gamma = 2.5$.

Figure 9: Impact of the amount of deposit data on the performance prediction error and the final average performance.

The third platform. To achieve the best collaboration effectiveness among different platforms, a moderate value of $\gamma$ is needed. On both datasets, the best average performance is achieved when $\gamma = 2.5$, which is consistent with our analysis in the methodology section.

4.6 Influence of Deposit Data Amount
To address RQ5, we compare the average platform performance and the average performance prediction errors\footnote{We report the errors under the uniform policy for consistency.} when using different percentages of deposit data, as shown in Fig. 9. We find that the performance prediction errors are usually large when the deposit datasets are too small. This is intuitive because accurate performance estimation usually needs a reasonable amount of data for model learning. Thus, the average performance is suboptimal if the deposit data is too sparse because the high prediction errors may impair the accuracy of policy optimization. However, the average performance also declines when the amount of deposit data is too large. This is because the privacy budget is depleted when a large amount of data is used as the deposit. Thus, using 5% of the total data as the deposit is suitable for our method.

5 CONCLUSION
In this paper, we study the platform collaboration problem in vertical federated learning under privacy constraints. We propose to use reciprocal collaborations to model the incentive mechanisms in VFL, and we further convert it into a multi-party game problem when considering constraints on platforms’ privacy budgets. To solve this game, we propose a FedGame approach that can generate effective collaboration policies. To evaluate platforms’ game rewards under different policies, we propose a performance estimation method that encourages each platform to offer a small amount of deposit data to participate in the federated learning of other platforms. Each platform independently trains and evaluates multiple models by involving different proportions of deposit data to obtain a regression function between its model performance and the amount of
different inter-platform data. Based on the estimated model performance, each platform participates in a negotiation process to bargain and optimize its collaboration policy. The final learning is conducted after the policies converge. Extensive experiments on the two datasets in different scenarios show that our approach can effectively facilitate the collaborations among platforms to better exploit multi-platform data under privacy restrictions.

However, our approach also has the following limitations. First, in our method, different platforms need to have certain trust, e.g., they do not collude with nor cheat each other. Second, in our method, we assume that all platforms have the consensus to use the same game solving strategy, which may require additional negotiation before starting the game. Third, our performance estimation method requires each platform to conduct multiple experiments, which may lead to additional computational and communication costs. In our future work, we plan to generalize our approach to scenarios with more relaxed assumptions about trust and game strategies. In addition, we will study more efficient methods to evaluate the value of inter-platform data.

REFERENCES

---

[1] Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. 2020. Vafl: a method of vertical asynchronous federated learning. arXiv preprint arXiv:2007.06891 (2020).

[2] Yao Chen and Agha Iqbal Ali. 2002. Output–input ratio analysis and DEA frontier. European Journal of Operational Research 142, 3 (2002), 476–479.

[3] Yang Liu, Xiong Zhang, and Libin Wang. 2020. Asymmetrical vertical federated learning. IEEE Access 8 (2020), 33–36.

[4] Zhiyuan Lyu, Xinyi Xu, Quan Wang, and Han Yu. 2020. Collaborative fairness in federated learning. In Federated Learning. Springer, 189–204.

[5] Sui Ma and Roshan Tourani. 2020. Predictive and causal implications of using shapley value for model interpretation. In Proceedings of the 2020 KDD Workshop on Causal Discovery. PMLR, 23–38.

[6] Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2018. Entity resolution and federated learning get a federated resolution. arXiv preprint arXiv:1803.04035 (2018).

[7] Tiantao Wang, Johannes Rausch, Cz Zhang, Ruoxi Jia, and Dawn Song. 2020. A principled approach to data valuation for federated learning. In Federated Learning. Springer, 153–167.

[8] Shaojun Feng, Dusit Niyato, Ping Wang, Dong In Kim, and Ying-Chang Liang. 2019. Game Theoretic Reasoning in Multi-Agent Coordination by Negotiation with a Trusted Third Party. In AGENTS. Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw (Eds.). ACM, 56–61.

[9] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. 2021. Feature valuation for participants in federated learning. In Proceedings of the 2021 ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 443–452.

[10] Shaohan Feng, Dusit Niyato, Fan Wang, Dongsok Kim, and Ying-Chang Liang. 2019. Joint service pricing and cooperative relay communication for federated learning. In 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, 815–820.

[11] Siwei Feng and Han Yu. 2020. Multi-participant multi-class vertical federated learning. arXiv preprint arXiv:2001.11154 (2020).

[12] Xiuqian Jin, Yawei Wang, and Junjie Wu. 2021. Data Valuation for Vertical Federated Learning: An Information-Theoretic Approach. arXiv preprint arXiv:2112.08364 (2021).

[13] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augusten, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018).

[14] Zirui Zhou, Lingyang Chu, Changxin Liu, Lanjun Wang, Jian Pei, and Yong Zhang. 2020. Game of Privacy: Towards Better Federated Platform Collaboration under Privacy Restriction. KDD '22, August 14–18, 2022, Washington D.C.
A SUPPLEMENTARY MATERIALS

A.1 Feature Field Partition

We list the feature fields on different simulated platforms as well as the features used as the prediction targets in Table 2. The feature orders are consistent with the original orders in the dataset. Note that the income feature is a binary variable that indicates whether the income of a person is higher than 50K.

| Platform | Feature       | Type       | Notes    |
|----------|---------------|------------|----------|
| 1        | age           | continuous |          |
|          | workclass     | categorical|          |
|          | final-weight  | continuous |          |
|          | education     | categorical|          |
|          | education-num | continuous | Removed  |
| 2        | marital-status| categorical|          |
|          | occupation    | categorical|          |
|          | relationship  | categorical|          |
|          | race          | categorical|          |
|          | gender        | categorical|          |
| 3        | capital-gain  | continuous |          |
|          | capital-loss  | continuous |          |
|          | hours-per-week| continuous |          |
|          | native-country| categorical|          |
|          | income        | categorical|          |

Table 2: The platform partition of feature fields.

A.2 Experimental Environment

Our experiments are conducted on a Linux machine with Ubuntu 16.04 operating system. The codes are written in Python 3.8. The deep learning algorithms are implemented by the Keras library 2.2.4 with Tensorflow 1.15 backend. On the Adult dataset, we only use CPU to run experiments. On the CTR dataset, we run experiment on a Nvidia Tesla V100 GPU with 32GB memory. Note that the negotiation process takes little computing resource and the game can be solved in seconds with common PCs.

A.3 Preprocessing

On the Adult dataset, we use one-hot encoding to map categorical features, and we normalize other numerical features. To fill missing values, we use the mean values for numerical features and the majority class for categorical features. On the CTR dataset, we use the NLTK tool to preprocess the texts. We use the word_tokenize function to convert the input texts into token sequences. We filter the low frequency tokens in search queries and browsed webpages using a frequency threshold of 10. The token embeddings of out-of-vocabulary words are filled with random vectors that have the same mean and co-variation values as other tokens.