2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)

关键应该是payoff calculation部分，但是找到在哪里描述的

# TradeFL: A Trading Mechanism for Cross-Silo Federated Learning

Shijing Yuan[1], Hongtao Lv[2], Hongze Liu[1], Chentao Wu[1], Song Guo[3], Zhi Liu[4], Hongyang Chen[5], Jie Li[1,6]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[2]School of Software & Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University
[3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[4]The University of Electro-Communications, Tokyo, Japan
[5]Research Center for Graph Computing, Zhejiang Lab, China
[6]MoE Key Lab of Artificial Intelligence, AI Institute Shanghai Jiao Tong University,
Yancheng Blockchain Research Institute, China
{2019ysj,seniordriver233,lijiecs,wuct}@sjtu.edu.cn,lht@sdu.edu.cn, song.guo@polyu.edu.hk,
liuzhi@uec.ac.jp, dr.h.chen@ieee.org.

***Abstract*—Cross-silo federated learning (CFL) is a distributed learning paradigm that allows organizations (e.g., financial or medical entities) to train a global model on siloed data. Recent studies on mechanisms designed for CFL, however, rarely jointly consider the potential inter-organizational competition and the lack of credibility between organizations, which may discourage organizational participation. In this paper, we investigate the problem of inter-organizational competition and credibility assurance. We propose a distributed trading mechanism, called *TradeFL*, to incentivize organizations to contribute data and computational resources through mutual trading among organizations. Technically, TradeFL characterizes the competition among organizations and compensates for their damage incurred by competition. TradeFL runs on distributed organizations and provides credibility guarantees for compensation through a customized smart contract[1]. We prove that the interaction among organizations that contribute resources to maximize personal payoffs is a weighted potential game. Then, we propose a centralized algorithm and a distributed algorithm to determine the optimal resource contribution. Simulation results and evaluations based on real-world datasets demonstrate that our scheme achieves higher social welfare, increases the amount of contributed data by up to 64%, and improves the accuracy of the global model by at most 23.2%.**

***Index Terms*—Trading mechanism, federated learning, potential game, smart-contract.**

## I. Introduction

Federated learning (FL) emerges as a promising technique that enables multiple clients with disjointed data and computational resources to collaboratively train a global model. In general, FL can be classified as cross-silo FL and cross-device FL [1]. The former consists of different organizations (e.g., financial or medical entities) [2] while the latter consists of mobile or IoT devices as clients. Cross-silo FL (CFL) is widely applied in the industry. For example, MELLODDY (the Machine Learning Ledger Orchestration for Drug Discovery) [3], a CFL project involving NVIDIA and several pharmaceutical companies, is designed for drug discovery. However, organizations may be reluctant to contribute data and computational resources to training. It is because those other competitors can also profit from their contributions if they are in competition [4]. Therefore, it is important to employ an effective mechanism to encourage organizations to contribute data when they are in competition.

Pertinent mechanisms [4]–[20] are mainly designed for cross-device FL, while there are few studies on the Incentive design for Cross-silo FL (ICFL). Recent works related to ICFL focus on resource contribution, (e.g., data or computational resources) [21], [22], social welfare maximization [23], and personalization [24]; see Sec. II for an overview. However, *most of them lack consideration of the co-existence of cooperation and competition among organizations, which is also known as "Coopetition"* [25], [26]. For instance, in MELLODDY, pharmaceutical companies with competitive relationships may resist engaging in collaborative training as their commercial share may be damaged by competitors using the global model to profit. Unfortunately, organizations involved in CFL are usually in an environment where competition is intense and frequent [27]. Furthermore, the aforementioned works either solely focus on maximizing computational resources or data contribution [21], [22], [28], or assume a predetermined bound of accuracy of the trained global model [29].

既有合作也有竞争

Why？有真实案例？

In this paper, we propose a trading mechanism, TradeFL, which aims to incentivize organizations to contribute data and computational resources through mutual trading among organizations. TradeFL operates within distributed organizations and models coopetition among organizations, while also compensating for organizational damage caused by cooperation. Importantly, compensation within TradeFL is facilitated directly through organizations, without relying on a central parameter server. Nevertheless, two challenges are confronted in accomplish-

[1]Illustration of the prototype: https://github.com/user10963.

2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-3986-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDCS57875.2023.00051

2575-8411/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDCS57875.2023.00051

Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

ing the above goals. 1) *It is difficult to determine exact compensation and resource contributions to improve social welfare.* Because organizations damaged by cooperation are reluctant to provide adequate resources without reasonable compensation. 2) *Organizations may exhibit dishonest behavior even with agreed-upon incentives or compensation.* For instance, malicious organizations may deny agreed compensation or deliberately reject proposals of the trading mechanism, leading to organizational disputes [30].

To address the above challenges, we first characterize the utility of organizations in coopetition, containing the revenue they gain from the global model and the competition damage they suffer. Notably, we do not assume any exact functional form between the amount of data contribution and accuracy of the trained global model, i.e., *data-accuracy function*, when calculating the revenue. Then, we propose a trading mechanism consisting of two components. To encourage organizations to contribute resources (both data and computational resources), the first component compensates for organizations' competition damage through payoff redistribution. The compensation is provided by organizations with small resource contributions to organizations with large resource contributions. Further, to prevent organizations from repudiation in mutual trading, the second component performs credible compensation by a smart-contract based prototype[1]. Because the immutability and traceability of smart contracts help to provide credibility guarantees for distributed trading mechanisms [30], [31]. Once the contract is signed by organizations, it can be automatically executed, guaranteeing undeniable and automated trading. Under the proposed TradeFL, we prove that the interaction among organizations that contribute resources to maximize personal payoffs is a *weighted potential game*. To obtain the resource contribution strategy with an approximate optimality guarantee, we propose a Centralized *Generalized Benders Decomposition* [32] (GBD)-based algorithm, i.e., CGBD. Despite theoretical approximate optimality guarantees, applying CGBD requires centralized control, which is challenging. Hence, we further propose a distributed algorithm based on *best response dynamics* [33] for organizations to autonomously decide their resource contribution strategies. We prove that our mechanism satisfies properties of *computational efficiency*, *individual rationality*, and *budget balance*.

Our main contributions are summarized as follows:

- *To our best knowledge, TradeFL is the first trading mechanism, which jointly considers the coopetition and credibility in CFL.* We conduct preliminary experiments to reveal the first and second derivative properties of the data-accuracy function; see Sec. III-C. In particular, TradeFL is conducive to practical adoption, since it does not rely on any specific form of the data-accuracy function.
- We propose a centralized algorithm to tackle the non-convexity of the potential function and analyze the optimality of the solution. We further propose a distributed algorithm with a lower complexity to achieve autonomous organizational decision-making.
- To provide undeniable and credible compensation for decentralized organizations, we implement a prototype[1] for the proposed mechanism by deploying a smart contract on an Ethereum [34] private chain.
- Simulation results show that our mechanism increases the amount of training data contribution by up to 64% and achieves the highest social welfare compared to baselines. Evaluations based on typical models and datasets demonstrate that our mechanism improves the accuracy of the global model by at most 23.2%.

**Paper Organization**. Section II summarizes the related works. Section III presents the pre-experiments and the trading mechanism. Section IV elaborates on coopetition game. Section V proposes the algorithms. Section VI conducts the evaluations. Section VII concludes this paper.

## II. Related work

Mechanisms designed for FL have emphasized the client-side contribution [5], [6], client-side privacy [7]–[9], fairness [10], [11], and personalization [24]. For instance, Lv *et al.* [5] and Xue *et al.* [6] proposed methods to evaluate client contributions without testing datasets and retraining, respectively. Other works such as IncenFL [7], IncenDCML [8], and IncenDPFL [9] explored the issue of compensation for privacy costs in incentive mechanisms. Studies [10], [11], and [24] designed mechanisms from the perspective of fairness and personalization, respectively. In addition, techniques such as the game theory [12], [13], contract theory [14], [15], auction [16], and blockchain [17]–[20] were applied to mechanism designs for FL. In particular, blockchain-based smart contracts are used to provide trustworthy assurance of incentives [31], [35], and to support automated trading mechanisms [30]. Distinctive from these works, we jointly explore the coopetition and credibility assurance in CFL.

Other related works [4], [27] studied competition in multi-party machine learning. Chen *et al.* [4] and Wu *et al.* [27] analyzed the phenomenon of competition in multi-party machine learning from the perspective of authenticity and market competition, respectively. These efforts take no consideration of reasonable compensation for the damage caused by competition. In addition, Tang *et al.* [21] developed an incentive mechanism based on the characteristics of public goods to encourage organizations to provide more computational resources, but ignored how much local data organizations should contribute to training. Huang [22] and Chen *et al.* [23] addressed the problem of inter-organizational free-riding through rational profit sharing and Multi-player Multi-action Zero-Determinant (MMZD) approach, respectively, but failed to take into account credibility assurance in CFL. Zhang *et al.* [29] considered the long-term selfish participation behavior of clients



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

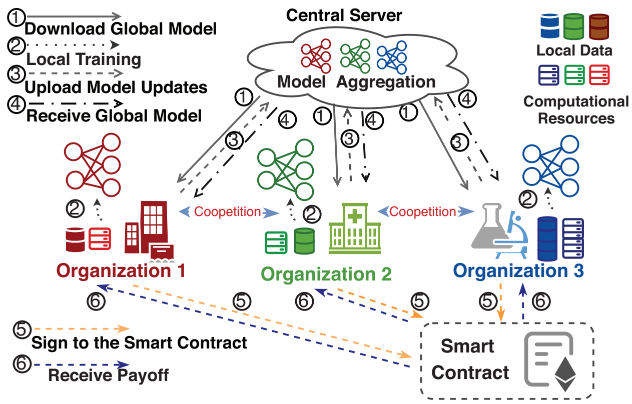



Fig. 1. CFL with coopetition (a scenario in which TradeFL operates), where organizations contribute data and computational resources to participate in training[2] with the aim of maximizing their payoffs.

to determine an optimal strategy of data contribution. However, the mechanism proposed in [29] depends on a given functional form of the data-precision function and is limited to specific model types.

Our trading mechanism is different since: 1) it incentive organizations to contribute resources through mutual trading between organizations without the central parameter server; 2) it extends prior works [4]–[24], [28]–[31] by jointly discussing the coopetition and credibility assurance in CFL; 3) it consists of a centralized algorithm that guarantees the optimality of resource contribution and a distributed algorithm suitable for application in real CFL scenarios; 4) it implements credible and automatic trading by developing a smart-contract based prototype.

## III. Proposed Trading Mechanism

### A. Overview

As shown in Fig. 1, we consider a practical cross-silo FL platform, where there is one central server and multiple organizations in coopetition. We denote the set of organizations as $\mathcal{O} = \{o_i\}_{i\in\mathcal{N}}$, where each organization has its own local data and computational resources. Let $\mathcal{S}_i$ be the local dataset of organization $o_i$, and $|\mathcal{S}_i|$ represents the number of data samples. The size of $o_i$'s local dataset and the total computational resources of $o_i$ are denoted as $s_i$ (bits) and $F_i^{(m)}$ (CPU cycles per second), respectively. Each organization $o_i$ can choose to participate in cross-silo FL or not. If $o_i$ chooses to take part in cross-silo FL, it needs to contribute at least more than $D_{min}s_i$ data, where $0 < D_{min} \leq 1$, while $o_i$ cannot obtain the global model if it does not choose to participate. By deciding on the proportion of data $d_i \in [D_{min}, 1]$, and computational resources $f_i \in [F_i^{(1)}, \ldots, F_i^{(m)}]$ to be contributed, organization $o_i$ aims to train a global model to maximize its payoffs. Once training[2] is complete, each $o_i$ signs the smart contract and receives the corresponding payoffs according to the proposed trading mechanism. The detailed designs are given in Section III-C - III-F.

### B. Training Process of Cross-silo FL

The training process of cross-silo FL can be divided into the following phases. 1) **Model Downloading.** Each organization $o_i$ downloads the global model from the central server. The average time spent by $o_i$ for downloading the global model is denoted as $T_i^{(1)}$. 2) **Local Training.** Organization $o_i$ selects a partial dataset with $d_i s_i$ data and $f_i$ computational resources to perform local training to obtain a locally updated model. Let $\boldsymbol{w}$ and $L_i(\boldsymbol{w})$ be the weight of the global model and the local loss function of $o_i$, respectively. Then $L_i(\boldsymbol{w})$ can be expressed as

$$L_i(\boldsymbol{w}) = \frac{1}{|\mathcal{S}_i|} \sum_{x_i^{(k)} \in \mathcal{S}_i} l\left(\boldsymbol{w}, x_i^{(k)}\right), \tag{1}$$

where $x_i^{(k)}$ is the $k$-th data sample in $\mathcal{S}_i$, and $l(\boldsymbol{w}, x_i^{(k)})$ is the loss function under $\boldsymbol{w}$ and $x_i^{(k)}$. The average time spent by $o_i$ on local training can be calculated as

$$T_i^{(2)}(d_i, f_i) = \frac{\eta_i d_i s_i}{f_i}, \tag{2}$$

where $\eta_i$ is the computational resources that required for $o_i$ to process each bit of local data. 3) **Model Uploading**. Organization $o_i$ uploads the updated local model $\boldsymbol{w}_i$ to the central server. The average time spent by $o_i$ on this process is $T_i^{(3)}$. 4) **Model Aggregation**. Finally, the central server updates the global model by using the FedAvg algorithm [1]. The loss function of the global model can be expressed as

$$L(\boldsymbol{w}) = \sum_{i=1}^{|\mathcal{N}|} \frac{d_i |\mathcal{S}_i|}{\sum_{k=1}^{|\mathcal{N}|} |\mathcal{S}_k|} L_i(\boldsymbol{w}). \tag{3}$$

To ensure the training efficiency, the above phases need to be completed within a specified deadline $\tau$ [1], i.e., $T_i^{(1)} + T_i^{(2)}(d_i, f_i) + T_i^{(3)} \leq \tau, \forall i \in \mathcal{N}$.

### C. Coopetition Model

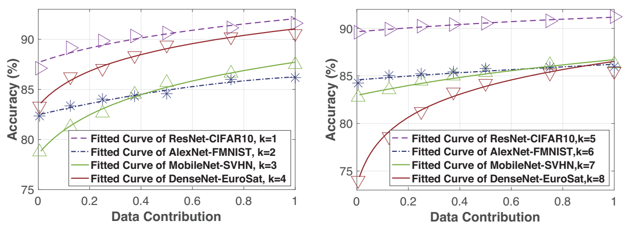
As mentioned earlier, there may be competition among organizations participating in cross-silo FL. In this subsection, we characterize this competition relationship by quantifying the revenue they obtain from the global model and the coopetition damage they suffer.

*1) **Revenue from the global model:*** Organization $o_i$ contributes its $d_i s_i$ data in expectation of a global model with good accuracy. Let $\boldsymbol{d_{-i}}$ be the data strategy set of all organizations except $o_i$, i.e., $\boldsymbol{d_{-i}} = \{d_j\}_{j\neq i, j\in\mathcal{N}}$. Denote $A\left(d_i, \boldsymbol{d}_{-i}\right)$ as $o_i$'s model accuracy loss. A smaller $A\left(d_i, \boldsymbol{d}_{-i}\right)$ indicates a better model accuracy [21], [29].

[2]TradeFL; see Section III-F; is applicable to both synchronous and asynchronous scenarios. It focuses on resource contribution without making assumptions about the asynchronicity of the training process.



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.




(a) $|\mathcal{N}|=2$, $i=1, \boldsymbol{-i}=2$ (b) $|\mathcal{N}|=4$, $i=2, \boldsymbol{-i}=\{1,3,4\}$

Fig. 2. Impact of $d_i$ on $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ with $d_{i'}=0.5, \forall d_{i'} \in \boldsymbol{d}_{-i}$. The number of data samples $\{|\mathcal{S}_i^k|\}_{i\in\mathcal{N}}$ is the same in the $k^{th}$ fitted curve, where $|\mathcal{S}_i^k| \in [2000, 20000]$.

By referring to [21], [29], the functional relationship between the data $d_i s_i$ contributed by $o_i$ and the accuracy of the trained global model, i.e., *data-accuracy function* $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})^3$, can be denoted as the difference between the accuracy loss $A(0)$ of the model without training and that with contributed data

$$\boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) = A(0) - A\left(d_i, \boldsymbol{d}_{-i}\right). \tag{4}$$

Accordingly, *the revenue of organization $o_i$ obtained from the global model can be expressed as* $p_i \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$, where $p_i > 0$ describes the profitability of $o_i$ [36], i.e., the revenue that $o_i$ earns from the unit performance of the global model. Notice that, with a strongly convex $L(\boldsymbol{w})$, $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) = \boldsymbol{P}(\sum_{i\in\mathcal{N}} d_i s_i)$ increases with $d_i s_i$, and the performance gain of $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ decreases when $d_i$ increases [21]. Based on the above observations and prior works [29], [37], we can assume that the first and second order derivative of $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})^4$ satisfies

$$\frac{\partial \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})}{\partial d_i} \geq 0, \frac{\partial^2 \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})}{\partial^2 d_i} \leq 0. \tag{5}$$

To further verify the impact of $d_i$ on $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$, we perform preliminary experiments[1] using classic models of ResNet-18, AlexNet, Densenet and MobileNet on datasets of CIFAR-10, FMNIST, SVHN and EuroSat [38]. The results in Fig. 2 show that $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ is increasing with respect to $d_i$, albeit at a muted growth rate, which verifies (5).

*2)* ***Damage incurred by coopetition****:* To quantify the coopetition damage that each $o_i$ suffers, we model coopetition among organizations as follows.

**Definition 1** (**Coopetiton**). *The co-existence of the co-operation in training and business competition between the participants of cross-silo FL is referred to as* "Coopetition" *[4], [27], [39]–[41].*

[3]In fact, the accuracy of the global model is not only related to $\{d_i, \boldsymbol{d}_{-i}\}$, but also related to training iterations $G$ and data quality [14]. We focus on the functional relationship between $\{d_i, \boldsymbol{d}_{-i}\}$ and $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$. The data quality and $G$ are taken as constants [28].

[4]The properties of $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ is investigated under the assumption that organizations' data are i.i.d [28], [29]. This is common in CFL, where organizations have data generated by labeling publicly available unlabeled datasets [28].

"Cooperation" refers to the participation of organizations in training with the expectation of profiting from the global model. As for the "competition", we use $\rho_{i,j} \in [0,1], i \neq j$ to denote this competition intensity between organization $o_i$ and organization $o_j$, which is usually measured by organizations' business metrics [42]. Note that $\rho_{i,j} = 0$ means there is no competition between $o_i$ and $o_j$. In fact, $\rho_{i,j}$ quantifies the degree of similarity among the services or products provided by $o_i$ and $o_j$ [26]. A larger $\rho_{i,j}$ means more intense competition between $o_i$ and $o_j$, i.e., the high similarity among the products or services offered by $o_i$ and $o_j$. For instance, $o_i$ and $o_j$ in the same industry contribute their local data to train a global model to improve the quality of their products. However, $o_i$'s market share may be damaged, because the quality of $o_j$'s product can be improved as well [27], especially when the competition intensity $\rho_{i,j}$ is large.

Clearly, $o_j$'s coopetition damage $\boldsymbol{D}_i(d_i, \boldsymbol{d}_{-i})$ is closely related to $\{d_i, \boldsymbol{d}_{-i}\}$ and $\rho_{i,j}$. Following prior works [26], [27], $\boldsymbol{D}_i(d_i, \boldsymbol{d}_{-i})$ can be measured by its competitors' profits. Hence, we can derive $\boldsymbol{D}_i(d_i, \boldsymbol{d}_{-i})$ in the following two steps. First, the profit $\varpi_j$ gained by $o_i$'s competitor $o_j$ because of $o_i$'s contribution to the global model can be calculated by

$$\varpi_j = p_j \left[\boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) - \boldsymbol{P}(0, \boldsymbol{d}_{-i})\right], j \neq i, \forall j \in \mathcal{N}, \tag{6}$$

where $P(d_i, \boldsymbol{d}_{-i}) - P(0, \boldsymbol{d}_{-i})$ is $o_i$'s contribution to the global model's accuracy performance, $p_j$ denotes the profitability of $o_j$. Second, $D_i(d_i, \boldsymbol{d}_{-i})$ can be calculated by the weighted sum of its competitors' profits

$$\boldsymbol{D}_i(d_i, \boldsymbol{d}_{-i}) = \sum_{j\in\mathcal{N}} \rho_{i,j} \varpi_j. \tag{7}$$

### D. Training Overhead

The training overhead of $o_i$ includes both computational and communication energy consumption. The former depends on the amount of contributed data and computational resources [21], which can be expressed as $E_i^{comp} = \kappa f_i^2 \eta_i d_i s_i$, where $\kappa$ [31] is the effective capacitance of the computational chipset. The latter can be given as $E_i^{comm} = E_{DL} T_i^{(1)} + E_{UL} T_i^{(3)}$, where $E_{DL}$ and $E_{UL}$ represent the energy consumption per unit time of the organizations during the model downloading and model uploading phases severally. Hence, the total energy consumption of $o_i$ can be calculated as

$$E_i = E_i^{comp} + E_i^{comm}. \tag{8}$$

### E. Formulation of the Proposed TradeFL

To incentivize organizations that are damaged by coopetition to contribute data and computational resources, in this subsection, we propose a trading mechanism considering coopetition compensation.

An effective way to compensate for the competition damage is *payoff redistribution*, i.e., the organization that



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

provides fewer resources should redistribute its payoffs to the organization that contributes more.

**Definition 2 (Payoff Redistribution).** *Let $\gamma$ be the incentive intensity, which describes the price of compensation per unit of contributed resource difference. The payoff redistribution that organization $o_i$ receives from its coopetitor $o_j$ is denoted by $r_{i,j}$, which can be expressed as*

$$r_{i,j} = \gamma \rho_{i,j} \left[ (d_i s_i + \lambda f_i) - (d_j s_j + \lambda f_j) \right], \tag{9}$$

*where $\lambda > 0$ is the parameter of uniform magnitude.*

Then the total payoff redistribution that $o_i$ obtains from its coopetitors can be calculated as

$$\boldsymbol{R}_i = \sum_{j \in \mathcal{N}} r_{i,j}. \tag{10}$$

Finally, *the payoff of organization $o_i$* that contributes $d_i s_i$ data and $f_i$ computational resources to training is

$$\begin{aligned} \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) &= p_i \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) - \varpi_e E_i - \boldsymbol{D}_i(d_i, \boldsymbol{d}_{-i}) + \boldsymbol{R}_i \\ &= p_i \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) - \varpi_e \left( \kappa f_i^2 d_i s_i + E_i^{(comm)} \right) \\ &- \sum_{j \in \mathcal{N}} \rho_{i,j} p_j \left[ \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) - \boldsymbol{P}(0, \boldsymbol{d}_{-i}) \right] + \sum_{j \in \mathcal{N}} r_{i,j}, \end{aligned} \tag{11}$$

where $\varpi_e$ is the weighting factor of the training overhead, $\pi_i = \{d_i, f_i\}$ is the resource contribution strategy of $o_i$ and $\boldsymbol{\pi_{-i}} = \{\pi_j\}_{j \neq i, j \in \mathcal{N}}$ is the strategy set of all the organizations except $o_i$, and thereby **social welfare** is $\sum_{i \in \mathcal{N}} \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i})$.

With payoff redistribution, the trading mechanism is designed with the goal of increasing data and computational resources invested by organizations to improve the accuracy of the global model while ensuring the key properties that are defined as follows.

**Definition 3 (Individual Rationality).** *Individual rationality refers to the fact that the payoff of each organization is non-negative, i.e., $\boldsymbol{\mathcal{C}_i}(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}) \geq 0$, where $\{\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\}$ is the strategy set of organizations under Nash equilibrium.*

**Definition 4 (Computational Efficiency).** *An incentive mechanism is computationally efficient if it can be conducted within a polynomial time.*

**Definition 5 (Budget Balance).** *Budget balance means that the proposed incentive mechanism can be sustained properly without increasing additional external incentives, i.e., $\sum_{i \in \mathcal{N}} \boldsymbol{R}_i = 0$.*

### *F. Prototype of the Proposed TradeFL*

As mentioned earlier, TradeFL enables payoff redistribution through mutual trading between organizations, without interacting with any third party. To ensure the credibility of this decentralized trading, we design a prototype[1] based on the smart contract. Specifically, the payoff redistribution $r_{i,j}$ is automatically executed through the proposed smart contract, thereby mitigating the potential for malicious behaviors to refuse to execute the payoff redistribution. Additionally, smart contracts ensure credible incentives by recording the results of the redistribution on blockchain. In the event of disputes between organizations, the recorded results can serve as a basis for arbitration and can be retroactively enforced.

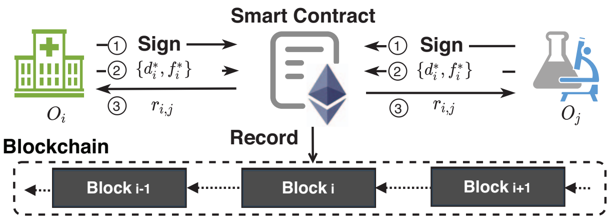



Fig. 3. Procedure of the proposed TradeFL based on smart-contract.

TABLE I
KEY FUNCTIONS IN THE PROPOSED SMART CONTRACT

| Function | Description |
|---|---|
| $depositSubmit()$ | Issue bonds to the contract |
| $contributionSubmit()$ | Submit contribution |
| $payoffCalculate()$ | Calculate the payoff |
| $payoffTransfer()$ | Perform payoff redistribution |
| $profileRecord()$ | Record the contribution profile |

As is shown in Fig. 3, the procedure of the smart contract consists of three steps. 1) First, each organization $o_i$ registers to the smart contract. Meanwhile, $o_i$ issues bonds (i.e., Ethereum virtual currency) by calling $depositSubmit()$ function[5] as their deposit. 2) After the training, $o_i$ reports its optimal contribution profile $\{d_i^*, f_i^*\}$[6] to the smart contract by calling $contributeSubmit()$ function. 3) The smart contract calculates $o_i$'s payoff through $payoffCalculte()$ function, and performs payoff redistribution through $payoffTransfer()$ function and refunds the margin proportionally. Note that $\{d_i^*, f_i^*\}$ and the corresponding payoff redistribution $r_{i,j}{}^*$ are obtained by the proposed algorithms in Section V.

## IV. COOPETITION GAME

### *A. Formulation of Coopetition Game*

Due to the great potential of game theory in analyzing the interactions among decentralized clients, we apply it to analyze organizations' optimal resource contribution. We model the interactions among organizations that contribute resources to maximize personal payoffs as a non-cooperative game. To be specific, each organization $o_i$ is a player who chooses its strategy $\pi_i = \{d_i, f_i\}$ with $d_i \in [D_{min}, 1]$ and $f_i \in [F_i^{(1)}, \ldots, F_i^{(m)}]$ to maximize its personal payoff $\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i})$. Then the coopetition game $\boldsymbol{\mathcal{G}}$ can be described as

[5] The functions developed in the smart contract are Application Binary Interfaces (ABI) in Ethereum [34].

[6] We assume that $\{d_i^*, f_i^*\}$ reported by organizations are truthful, which can be verified through the Trusted Execution Environments (TEE) proposed in [43]. Verification for the correctness of organizations' reports is beyond the scope of this paper.



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

- *Players*: all organizations $\{o_i\}_{i\in\mathcal{N}}$.
- *Strategy set*: data and computational resources $\{\pi_i = \{d_i, f_i\}\}_{i\in\mathcal{N}}$ contributed by organizations.
- *Payoff function*: $\{\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i})\}_{i\in\mathcal{N}}$ defined in Eq. (11).

**Definition 6 (Nash Equilibrium)**. *A Nash equilibrium (NE) of $\mathcal{G}$ is a strategy profile $\pi^* = \{\pi_i\}_{i\in\mathcal{N}}$ in which no $o_i$ can maximize its payoff by unilaterally deviating its own strategy, e.g.,*

$$\boldsymbol{\mathcal{C}_i}(\pi_i^*, \boldsymbol{\pi}_{-i}^*) \geq \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}^*), \quad \forall i \in \mathcal{N}. \tag{12}$$

Under the above definition and given other organizations' strategies $\boldsymbol{\pi}_{-i}$, each organization $o_i$ decides its resource strategy $\pi_i = \{d_i, f_i\}$ to maximize its payoff

$$\begin{aligned} &\max_{\pi_i} \quad \boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) \\ &C_i^{(1)} : d_i \in [D_{min}, 1], \\ &C_i^{(2)} : f_i \in [F_i^{(1)}, \ldots, F_i^{(m)}], \\ &C_i^{(3)} : T_i^{(1)} + T_i^{(2)}(d_i, f_i) + T_i^{(3)} \leq \tau, \end{aligned} \tag{13}$$

where $C_i^{(1)}$ and $C_i^{(2)}$ ensure the validity of the resources strategy, and $C_i^{(3)}$ means that the average time spent by $o_i$ on training should not exceed the given deadline $\tau$.

### B. Existence of the Nash Equilibrium

With the coopetition game $\boldsymbol{\mathcal{G}}$, we investigate the existence of the NE of it. To proceed, we begin by introducing the concept of weighted potential game.

**Definition 8 (Weighted Potential Game)**. *A game $\mathcal{G}$ is a weighted potential game if it holfd a potential function $\boldsymbol{U}(\pi_i, \boldsymbol{\pi}_{-i})$ [33] that satisfies the following equation for all $i \in \mathcal{N}$*

$$z_i \left[\boldsymbol{U}(\pi_i, \boldsymbol{\pi}_{-i}) - \boldsymbol{U}(\pi_i', \boldsymbol{\pi}_{-i})\right] = \boldsymbol{\mathcal{C}}_i(\pi_i, \boldsymbol{\pi}_{-i}) - \boldsymbol{\mathcal{C}}_i(\pi_i', \boldsymbol{\pi}_{-i}), \tag{14}$$

*where $z_i$ is the non-negative weighted factor.*

According to the potential game theory [33], in weighted potential games, the NE exists and the change in each $o_i$'s $\boldsymbol{\mathcal{C}}_i(\pi_i, \boldsymbol{\pi}_{-i})$ due to its strategy deviation is equal to the change in $\boldsymbol{U}(\pi_i, \boldsymbol{\pi}_{-i})$ but scaled by a weighting factor $z_i$.

**Theorem 1 ($\mathcal{G}$ is A Weighted Potential Game).** *The coopetition game $\boldsymbol{\mathcal{G}}$ is a weighted potential game with the potential function that is given in Eq. (15) and hence the pure strategy NE of $\boldsymbol{\mathcal{G}}$ is guaranteed to exist.*

$$\boldsymbol{U}(\boldsymbol{\pi}) = \boldsymbol{P}(\Omega) - \sum_{i\in\mathcal{N}} \left[\frac{\varpi_e \kappa f_i^2 \eta_i d_i s_i}{z_i} - \sum_{j\in\mathcal{N}} \frac{r_{i,j}}{z_i}\right], \tag{15}$$

*where $\Omega = \sum_{i\in\mathcal{N}} d_i s_i$ and $z_i = p_i - \sum_{j\in\mathcal{N}} \rho_{i,j} p_j > 0$. $\rho_{i,j}$ is mapped to a small number to ensure that $z_i > 0$.*

***Proof***. We consider that $o_i$ changes its strategy from $\pi_i = \{d_i, f_i\}$ to $\pi_i' = \{d_i', f_i'\}$ while the strategy set $\boldsymbol{\pi}_{-i}$ of other organizations $\boldsymbol{o}_{-i}$ remains unchanged, then the potential function can be written as $\boldsymbol{U}(\boldsymbol{\pi}')$.

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{\pi}') = {} & \boldsymbol{P}(d_i', \boldsymbol{d}_{-i}) - \sum_{j\in\mathcal{N}, j\neq i} \frac{\varpi_e \kappa f_j^2 d_j s_j}{z_i} - \frac{\varpi_e \kappa f_i'^2 d_i' s_i}{z_i} \\ & + \sum_{j\in\mathcal{N}} \frac{r_{i,j}(\pi_i', \pi_j)}{z_i} + \sum_{j\in\mathcal{N}, j\neq i} \sum_{k\in\mathcal{N}} \frac{r_{j,k}(\pi_j, \pi_k)}{z_i}. \end{aligned} \tag{16}$$

Subtracting Eq. (16) from Eq. (15), we have

$$\boldsymbol{U}(\boldsymbol{\pi}) - \boldsymbol{U}(\boldsymbol{\pi}') = \frac{1}{z_i}\left[\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i}) - \boldsymbol{\mathcal{C}_i}(\pi_i', \boldsymbol{\pi}_{-i})\right]. \tag{17}$$

Clearly, Theorem 1 holds true regardless of whether $\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ is a convex function, as long as it satisfies Eq. (5), which completes the proof of Theorem 1. □

According to [33, Theorem 2.4], the solution of the potential function is the NE of the potential game. Therefore, in the next section, we propose two algorithms to achieve the NE of $\boldsymbol{\mathcal{G}}$ by solving the solution of (15).

## V. The Proposed Algorithms to Achieve the Nash Equilibrium

### A. Finding the Global Solution of the Potential Function

As aforementioned, the NE of the coopetition game can be achieved by finding the global solution of the potential function (15). Therefore, we formulate the optimization problem of finding the global solution of (15) as

$$\begin{aligned} &\max_{\boldsymbol{\pi}} \quad \boldsymbol{U}(\boldsymbol{\pi}) \\ &\text{s.t. } C_i^{(1)}, C_i^{(2)}, C_i^{(3)}, \forall i \in \mathcal{N}, \end{aligned} \tag{18}$$

where $\boldsymbol{\pi} = \{\pi_i = \{d_i, f_i\}\}_{i\in\mathcal{N}}$ is the strategy set of all organizations. For any Eq. (4) satisfying (5), the problem (18) is a mixed integer nonlinear programming problem (MINLP), and the existence of $f_i^2$ makes it non-convex. Fortunately, it can be found that if $f_i$ is fixed, then (18) can be transformed into a convex problem. Meanwhile, the projection of (18) onto $f_i$ becomes an integer programming problem, whose solution can be obtained by using the duality theory and the relaxation method [32]. Based on the above observations, we propose two GBD-based algorithms to solve (18) since GBD [32] is an iterative algorithm widely applied in solving MNILP problems.

### B. Proposed Centralized Algorithm *CGBD*

To solve problem (18) with guaranteed approximate optimality, we propose a centralized algorithm based on GBD, *i.e.,* **CGBD**. *Note that our CGBD algorithm does not rely on any exact functional form of Eq. (4).* With **CGBD**, (18) can be decomposed into a *primal problem* and a *master problem* as given in (19) and (23), respectively, which can be iteratively solved. Let $k$ be the $k^{th}$ iteration. Then the *primal problem* can be expressed as

$$\begin{aligned} &\textbf{Primal Problem}: \min_{\boldsymbol{d}} -\boldsymbol{U}(\boldsymbol{d}, \boldsymbol{f}^{(k-1)}) \\ &\text{s.t. } d_i \in \mathcal{X}_i, \forall i \in \mathcal{N}, \\ &\qquad T_i^{(1)} + T_i^{(2)}(d_i, f_i^{(k-1)}) + T_i^{(3)} \leq \tau, \forall i \in \mathcal{N}, \end{aligned} \tag{19}$$



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

where $\mathcal{X}_i \triangleq [D_{min}, 1]$. According to **Lemma 1**, (19) is a convex problem. Solving (19) can yield the optimal value $-\boldsymbol{U}(\boldsymbol{d}^{(k)}, \boldsymbol{f}^{(k-1)})$ and Lagrange multiplier vector $\boldsymbol{u}^{(k)}$. Then the Lagrange function of (19) can be given as

$$\boldsymbol{\mathcal{L}}^{*}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}, \boldsymbol{u}^{(k)}\right) = \boldsymbol{U}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right) + \boldsymbol{u}^{(k)^{T}} \boldsymbol{G}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right), \tag{20}$$

where $\boldsymbol{G}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right) = \{T_i^{(1)} + T_i^{(2)}(d_i^{(k)}, f_i) + T_i^{(3)} - \tau\}_{i \in \mathcal{N}}$. However, not all of $\boldsymbol{f}^{(k-1)}$ can lead to a feasible *primal problem*. If (19) is infeasible with some specific values of $\boldsymbol{f}^{(k-1)}$, we turn to the following problem instead

$$\begin{aligned} \min_{\{\boldsymbol{d}, \zeta\}} \quad & \zeta \\ \text{s.t. } & \boldsymbol{G}\left(\boldsymbol{d}, \boldsymbol{f}^{(k-1)}\right) \leq \zeta, \boldsymbol{d} \in \boldsymbol{\mathcal{X}}, \zeta \geq 0, \end{aligned} \tag{21}$$

where $\boldsymbol{\mathcal{X}} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_{|\mathcal{N}|}, d_i \in \mathcal{X}_i$. Problem (21) is known as the feasibility-check problem [32], whose result reflects the feasibility of (19). Specifically, $\zeta > 0$ indicates that (19) is infeasible; otherwise, it is feasible. Also, the Lagrange multiplier vector $\boldsymbol{\lambda}^{(k)}$ of (21) can be obtained by solving (21), and thereby the Lagrange function of (21) is

$$\boldsymbol{\mathcal{L}}_{*}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}, \boldsymbol{\lambda}^{(k)}\right) = \boldsymbol{\lambda}^{(k)^{T}}\left(\boldsymbol{G}\left(\boldsymbol{d}^{(k)}, \boldsymbol{f}\right) - \zeta\right). \tag{22}$$

Let $\mathcal{V}_{fea}^{(k)}$ and $\mathcal{I}_{inf}^{(k)}$ severally be the set of feasible and infeasible iteration indexes. Then the *master problem* can be formulated as

$$\begin{aligned} \textbf{Master Problem}: \min_{\{\boldsymbol{f}, \varphi\}} \quad & \varphi \\ \text{s.t. } \varphi \geq \boldsymbol{\mathcal{L}}^{*}&\left(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{u}_v^{(k)}\right), \forall v \in \mathcal{V}_{fea}^{(k)}, \boldsymbol{u}^{(k)} \succeq \boldsymbol{0}, \\ 0 \geq \boldsymbol{\mathcal{L}}_{*}&\left(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{\lambda}_v^{(k)}\right), \forall v \in \mathcal{I}_{inf}^{(k)}, \boldsymbol{f} \in \boldsymbol{\mathcal{F}}, \end{aligned} \tag{23}$$

where $\boldsymbol{\mathcal{F}} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{|\mathcal{N}|}, f_i \in \mathcal{F}_i, \mathcal{F}_i \triangleq [F_i^{(1)}, \ldots, F_i^{(m)}]$. Consequently, the global solution of (18) can be obtained by iteratively solving the primal problem (19) and the master problem (23) until the upper bound $UB^{(k)}$ and lower bound $LB^{(k)}$ satisfy the convergence condition, i.e., $UB^{(k)} - LB^{(k)} \leq \epsilon$, where $\epsilon$ is the accuracy tolerance. The whole procedure of **GCBD** is demonstrated in **Algorithm 1**. **Lemma 1** analyzes the convexity of (19).

**Lemma 1 (Convexity Analysis).** *The primal problem (19) is a convex problem with any* $\boldsymbol{f}^{(k-1)} \in \boldsymbol{\mathcal{F}}$.

***Proof*.** According to (5), (15) and (19), the second-order derivative of $-\boldsymbol{U}(\boldsymbol{d}, \boldsymbol{f}^{(k-1)})$ with respect to $d_i$ is $-\nabla_{d_i}^2 \boldsymbol{U} = -\nabla_{d_i}^2 \boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$. Then we have $-\nabla_{d_i}^2 \boldsymbol{U} \geq 0$ since $-\nabla_{d_i}^2 \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) \geq 0$. Thus we draw the conclusion that the primal problem (19) is a convex problem with linear differentiable constraints. □

Based on Lemma 1, we obtain $\boldsymbol{d}^{(k)}$ and $\boldsymbol{u}^{(k)}$ whose objective value $-\boldsymbol{U}(\boldsymbol{d}^{(k)}, \boldsymbol{f}^{(k-1)})$ is within $\delta$ of the optimal solution by *interior-point* (IP) method [44], where $\delta > 0$ is the accuracy tolerance. Since (19) and (21) have similar structures, (21) can also be solved by the IP method. Details of the IP method [44] are omitted due to the length limitation of this paper. Moreover, the master problem

**Algorithm 1: CGBD** algorithm for solving (18).

**Input:** Iteration index, k=0, the tolerance $\epsilon$, $UB^{(0)} = \infty, LB^{(0)} = -\infty$, and $\boldsymbol{f}^{(0)} \in \boldsymbol{\mathcal{F}}$.

**Output:** The global solution $\boldsymbol{\pi}^{NE} = \{\boldsymbol{d}^{*}, \boldsymbol{f}^{*}\}$.

**while** $UB^{(k)} - LB^{(k)} > \epsilon$ *and* $k < K$ **do**
- $k \leftarrow k + 1$;
- **if** *the primal problem (19) is feasible* **then**
  - Solve (19) with $\boldsymbol{f}^{(k-1)}$ given and obtain $\boldsymbol{d}_v^{(k)}$, $\boldsymbol{u}_v^{(k)}$, $\boldsymbol{\mathcal{L}}^{*}(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{u}_v^{(k)})$;
- **else**
  - Solve problem (21) and obtain $\boldsymbol{d}_v^{(k)}$, $\boldsymbol{u}_v^{(k)}$, and $\boldsymbol{\mathcal{L}}_{*}\left(\boldsymbol{d}_v^{(k)}, \boldsymbol{f}, \boldsymbol{\lambda}_v^{(k)}\right)$;
- $UB^{(k)} \leftarrow min\{UB^{(k-1)}, \boldsymbol{U}(\boldsymbol{d}^{(k)}, \boldsymbol{f}^{(k-1)})\}$;
- Solve the master problem (23), obtain $\boldsymbol{f}^{(k)}$, $\varphi^{*}$;
- Update the lower bound $LB^{(k)} \leftarrow \varphi^{*}$;

Return the global solution $\boldsymbol{\pi}^{NE} = \{\boldsymbol{d}^{*}, \boldsymbol{f}^{*}\}$;

(23) is a mixed integer programming (MIP) problem. Since we do not focus on solving the MIP problem, the traversal method is applied only, i.e., the solution of (23) is obtained by exhaustively enumerating the feasible values of $\boldsymbol{f}^{(k)}$.

### C. *Performance Analysis for **CGBD** Algorithm*

We have the following lemmas of convergence, optimality, and computational complexity for **CGBD**.

**Lemma 2 (Finite Convergence)**. *The proposed CGBD algorithm terminates in a finite number of steps for any given $\epsilon > 0$ even when $\epsilon = 0$.*

***Proof*.** According to [32, Theorem 2.4], the finite termination of our CGBD algorithm adheres to the fact that no $\boldsymbol{f}$ can repeat itself in the solution of (23). Since the domain of $\boldsymbol{f}$ is a finite discrete set, the upper bound $UB^{(k)}$ and the lower bound $LB^{(k)}$ of iteration satisfy the convergence criterion $UB^{(k)} - LB^{(k)} \rightarrow \epsilon$ even when $\epsilon = 0$. □

**Lemma 3 (($\delta$+$\epsilon$)-Optimality)**. *The solution generated by Algo. 1 is a $(\delta + \epsilon)$-optimal solution of (18).*

***Proof*.** Based on [32, Theorem 2.1], $\{\boldsymbol{d}^{*}, \boldsymbol{f}^{*}\}$ is $(\delta + \epsilon)$-optimal in (18) since $\boldsymbol{d}^{*}$ obtained by the IP method is with $\delta$-optimality [44] in (19) and $\boldsymbol{f}^{*}$ is $\epsilon$-optimal [32] in (23). That is, $\{\boldsymbol{d}^{*}, \boldsymbol{f}^{*}\}$ generated by Algo. 1 is with ($\delta$+$\epsilon$)-optimality of the global optimum of (18). □ **Lemma 4 (Computational Complexity)**. *Computational complexity of Algo. 1 is $\mathcal{O}(Im^{|\mathcal{N}|})$, where $I \leq K$.*

***Proof*.** In Algo. 1, the computational complexity of the IP method [44] in solving the primal problem (19) is $O\left(\sqrt{|\mathcal{N}|} log(\frac{|\mathcal{N}|}{\delta})\right)$. The computational complexity of our traversal method in solving the master problem (23) is $\mathcal{O}(m^{|\mathcal{N}|})$. The total computational complexity of Algo. 1 is $\mathcal{O}(I(\sqrt{|\mathcal{N}|} log(\frac{|\mathcal{N}|}{\delta}) + m^{|\mathcal{N}|})) = \mathcal{O}(Im^{|\mathcal{N}|})$, where $I$ is the number of iterations until the algorithm converges. □



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

**Algorithm 2: DBR** algorithm to achieve the NE.

**Input:** $\{\pi_i^{(0)}\}_{i \in \mathcal{N}}$, where $d_i^{(0)} = D_{min}, f_i^{(0)} = F_i^{(m)}$, and the decision slot $t \leftarrow 0$.

**Output:** The strategy set $\boldsymbol{\pi}^{NE}$.

**while** $t < H$ **do**
  $t \leftarrow t + 1$ ;
  Compute each $o_i's$ best response $\pi_i'$ by GBD
  **if** $\pi_i' \neq \pi_i^{[t-1]}$ **then**
    Update $o_i's$ strategy $\pi_i^{[t]} \leftarrow \pi_i'$;
  **if** *No organization changes its strategy* **then**
    Break;
Return $\boldsymbol{\pi}^{NE} = \{\boldsymbol{d}^*, \boldsymbol{f}^*\}$;

### D. Proposed Distributed Algorithm ***DBR***

Although CGBD provides a $(\delta + \epsilon)$-optimal solution, performing CGBD requires a central controller, which is challenging for decentralized organizations. To achieve autonomous organizational decision-making in real CFL scenarios, we propose a distributed algorithm based on *best response dynamics* [33], i.e., **DBR**. Next, we introduce the definition of *best response*.

**Definition 9 (Best Response)**. *The best response (BR) of organization $o_i$ in consideration of the strategy set $\boldsymbol{\pi}_{-i}$ of other organizations, constitutes the strategy that $o_i$ should employ to maximize its own payoff.*

$$\pi_i' = \underset{d_i \in \mathcal{X}_i, f_i \in \mathcal{F}_i}{\arg\max} \quad \boldsymbol{\mathcal{C}}_i\left(\pi_i, \boldsymbol{\pi}_{-i}\right). \tag{24}$$

Similar to solving (18), $o_i$'s best response can be obtained by the proposed GBD-based algorithm since (24) has a similar structure to (18). According to [33], the NE of potential game can be reached by performing iterative best responses of players. Therefore, the NE of the coopetition game can be achieved after finite updates of organizations' best response. The whole procedure of the **DBR** algorithm is given in **Algorithm 2**.

*Notably, DBR allows organizations to autonomously determine their resource contributions $\boldsymbol{\pi}^{NE}$ in a distributed manner, without the need for interaction with a central parameter server. Even when $\boldsymbol{P}(d_i, \boldsymbol{d}{-}i)$ is non-convex, DBR can still obtain $\boldsymbol{\pi}^{NE}$ because the coopetition game Eq. (13) is still a potential game ($\boldsymbol{P}(d_i, \boldsymbol{d}_{-i})$ that satisfies $\boldsymbol{P}(d_i', \boldsymbol{d}_{-i}) - \boldsymbol{P}(d_i, \boldsymbol{d}_{-i}) \geq 0, \forall d_i' \geq d_i, d_i', d_i \in \mathbb{R}$).*

Given that the convergence of BR-based algorithms has been demonstrated in [33], our analysis focuses solely on the computational complexity of Algo. 2. It is straightforward that the complexity of Algo. 2 is reduced from $\mathcal{O}(Im^{|\mathcal{N}|})$ to $\mathcal{O}(TL|\mathcal{N}|m)$, where $T \leq H$ and $L \leq I$.

**Theorem 2 (Properties of TradeFL).** *The proposed TradeFL satisfies desirable properties, including individual rationality (IR), budget balance (BB), and computational efficiency (CE).*

TABLE II
EXPERIMENTAL PARAMETERS

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $\lvert\mathcal{N}\rvert$ | 10/0.01 | $p_i$ | [500,2500] |
| $s_i$ | [15,25]*$10^9$ | $\lvert\mathcal{S}_i\rvert$ | [1000,2000] |
| $\kappa$ | $10^{-27}$ | $F_i^{(m)}$ | $3-5GHz$ |

***Proof***. Let $\widetilde{\pi}_i = \{D_{min}, f_i\}, i \in \mathcal{N}$. Given $p_i, \rho_{i,j}$, it is easy to find a $\gamma$ such that $\mathcal{C}_i(\widetilde{\pi}_i, \boldsymbol{\pi}_{-i})$ is non-negative. Under the NE of $\mathcal{G}$, we have $\mathcal{C}_i(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}) \geq \mathcal{C}_i(\widetilde{\pi}_i, \boldsymbol{\pi}_{-i}) \geq 0$, and hence IR holds. According to Eq. (9), we have $\sum_{i \in \mathcal{N}} \boldsymbol{R}_i = 0$, so that the proposed incentive mechanism is BB without external incentives. The computational complexity of Algo. 2 is $\mathcal{O}(TL|\mathcal{N}|m)$, and thus the incentive mechanism can be conducted within a polynomial time. $\square$

## VI. Experiments and Results

We first conduct simulations to analyze the performance of TradeFL in terms of dynamics of the coopetition game, social welfare, and coopetition damage. Additionally, we highlight the impact of key parameters on social welfare. Then we conduct real evaluations for TradeFL using typical models that mentioned in Sec. III-C. In order not to be restricted to specific model-dataset combinations, we adopt the general accuracy loss function[7] proposed in [21], [29], [37] in our simulations. The key parameters used in the simulations are shown in Table II.

**Prototype**. We design a smart contract on an Ethereum private blockchain to implement the prototype of TradeFL. The proposed prototype runs on Ubuntu 18.07 (Intel Xeon 5222 CPU@3.80GHz and 128GB Memory). The smart contract is deployed in 41 lines of Solidity (v0.5.17) [34] language. Web3 API [34] is utilized for data interaction among organizations and the smart contract when calling contract functions (see Table. I).

**Baselines**. Since the trading mechanism with the perspective of coopetition has rarely been considered in other works, the existing works can hardly be adopted as comparison baselines without modification. Hence, we consider the following methods as baselines.

- **DBR Without Payoff Redistribution (WPR)**. In this case, organizations derive payoff solely from the global model [29], that is, Eq. (10) is not included in $\boldsymbol{\mathcal{C}_i}(\pi_i, \boldsymbol{\pi}_{-i})$.
- **DBR with Greedy Computation Allocation (GCA)**. Organizations contribute computational resources greedily in proportion to the amount of data they provide, i.e., $f_i = kd_i$, where $k$ is a constant.
- **Finite Improvement Property (FIP) based Scheme**. The NE can be achieved by performing

[7] Following [21], [29], [37], $o_i's$ model accuracy loss can be bounded as $A\left(d_i, \boldsymbol{d}_{-i}\right) = \frac{1}{\sqrt{\left(d_i s_i + \sum_{j \in \mathcal{N}, j \neq i} d_j s_j\right) G}} + \frac{1}{G}$, which we use for simulations, where $G$ is the number of training epochs.



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

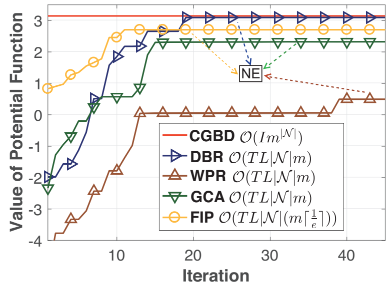



Fig. 4. Potential Function Dynamics under Various Schemes.

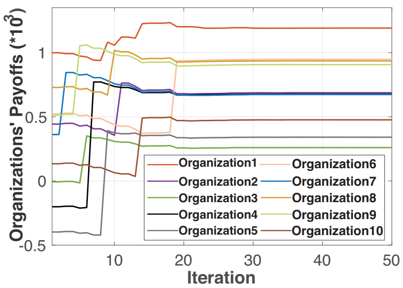



Fig. 5. Payoff Dynamics of Organizations under DBR.

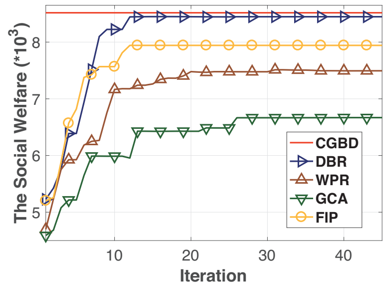



Fig. 6. Social Welfare under Different Schemes.

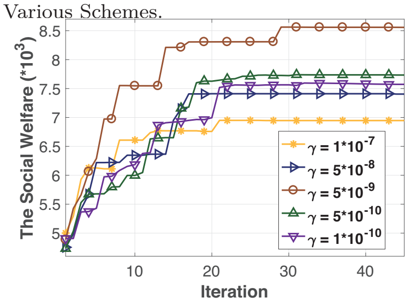



Fig. 7. Impact of $\gamma$ on Social Welfare under DBR.

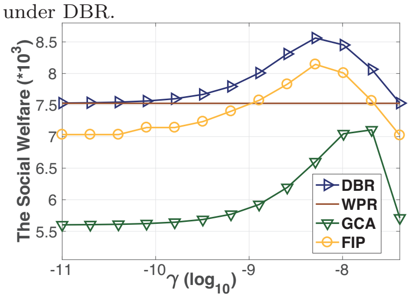



Fig. 8. Comparing Social Welfare under Various Schemes with respect to $\gamma$.

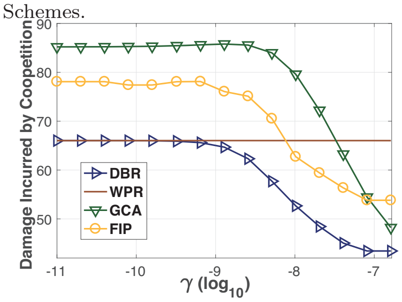



Fig. 9. Coopetition Damage under Different Schemes with respect to $\gamma$.

finite updates of the organizations' strategy set based on FIP [33] of potential game, i.e., $\{\widehat{\pi}_i = \{\widehat{d}_i, f_i\}\}_{i\in\mathcal{N}}$, where $\widehat{d}_i \in [e, 2e, \ldots, 1], e \in [D_{min}, 1]$.

- **Theoretically Optimal Scheme (TOS)**. Each organization provides all local data and computational resources, ignoring both the constraint of training time and the coopetition damage.

**Fig. 4 shows the dynamics of the value of the potential function**. We observe that although all schemes converge to the NE of the coopetition game and CGBD achieves the largest value of the potential function. Besides, the gap between CGBD and DBR is rather small. Given the fact that compared to baselines, the proposed mechanism not only provides reasonable coopetition compensation, but also customizes more efficient resource contribution strategies for each organization, giving rise to a larger value of the potential function.

**Fig. 5 depicts the dynamics of organizations' payoffs $\mathcal{C}_i(\pi_i, \boldsymbol{\pi}_{-i})$ under DBR.** With each iteration, DBR allows each organization to compute its best response to determine the resource contribution autonomously. After 25 iterations, $\mathcal{C}_i(\pi_i, \boldsymbol{\pi}_{-i})$ gradually converges to the NE. It is worth noting that, DBR achieves the NE in a distributed way, which is applicable to practical CFL scenarios.

**Fig. 6 illustrates social welfare under different schemes**. As is shown in Fig. 6, CGBD attains the highest social welfare, followed by DBR. The reason is that CGBD and DBR direct the organization's resource contribution strategy in the optimal direction through payoff redistribution, thus achieving higher social welfare. Compared with DBR and CGBD, WPR does not provide compensation for organizations' coopetition damage. This lack of compensation discourages organizations from contributing additional data $\boldsymbol{d}$ and computational resources $\boldsymbol{f}$, subsequently leading to subpar global model performance and decreased social welfare. Moreover, solving $\boldsymbol{d}$ and $\boldsymbol{f}$ under FIP on the limited strategy space can only obtain a compromise solution. Although GCA can satisfy the training time constraint, it always decides $\boldsymbol{f}$ greedily according to $\boldsymbol{d}$, leading to a sub-optimal solution.

**Figs. 7 - 9 investigate the impact of the incentive intensity $\gamma$ (defined in Eq. (9))**. The results in Figs. 7 - 8 illustrate how social welfare varies with $\gamma$. *The results highlight that increasing $\gamma$ does not always improve social welfare.* This is because a large $\gamma$ also leads to a high training overhead, despite the fact that a large $\gamma$ setting can incentive organizations to contribute more resources to improve the accuracy of the global model and thus obtain a higher payoff. As depicted in Fig. 7, social welfare drops when $\gamma = 5 \times 10^{-8}$ and $1 \times 10^{-7}$. Fig. 9 shows the impact of $\gamma$ on the coopetition damage varies under different schemes. Due to the marginal effect of data contribution, the coopetition damage decreases as $\gamma$ increases for all schemes except WPR. *In summary, DBR achieves the highest social welfare and lowest coopetition damage as compared to baselines* benefiting from payoff redistribution and the more precise resource contribution.

**Figs. 10 - 11 study the impact of $\gamma$, the weighting factor of the training overhead $\varpi_e$, and the mean of the competition intensity $\rho_{i,j}$ on social welfare**. In this simulation, $\rho_{i,j}$ are randomly generated using a normal distribution with mean $\mu$ and standard deviation



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

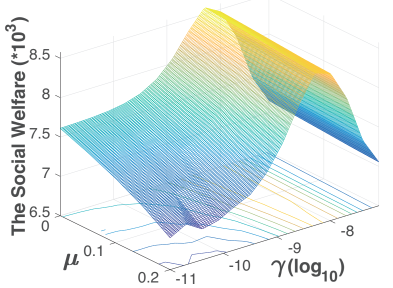


Fig. 10. Combined Impact of $\gamma$ and $\mu$ on Social Welfare under DBR.

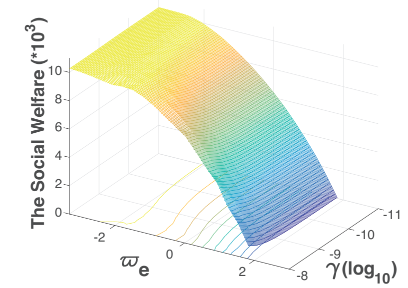


Fig. 11. Combined Impact of $\gamma$ and $\varpi_e$ on Social Welfare under DBR.

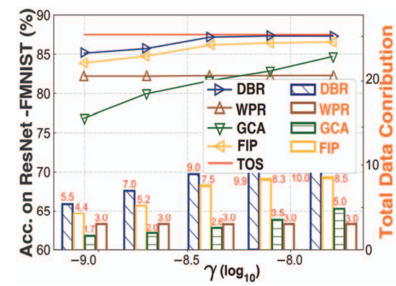


Fig. 12. The Trained Global Model's Accuracy and the Total Data Contribution across Different $\gamma$.

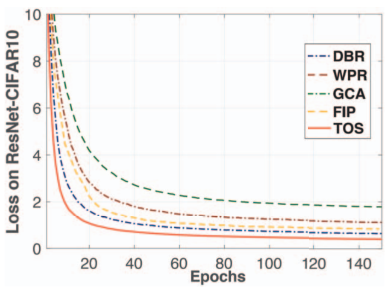


Fig. 13. Loss. ResNet-CIFAR-10.

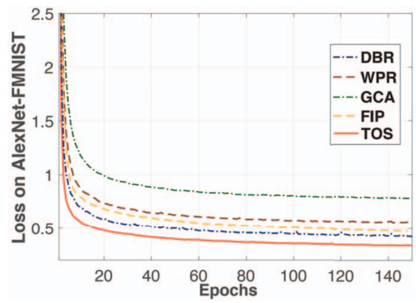


Fig. 14. Loss. AlexNet-FMINIST.

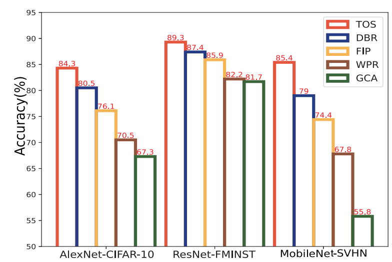


Fig. 15. The Trained Global Model's Accuracy accross Different Schemes.

$\sigma^2 = \frac{\mu}{5}$, where $\rho_{i,j} \sim N(\mu, \sigma^2)$. Fig. 10 demonstrate that with the rise of $\gamma$, social welfare surges to 8582.7 when $\gamma^* = 5.12 \times 10^{-9}$ and then drops to 6891.7, as $\gamma$'s impact on social welfare is non-monotonic. When $\gamma \geq \gamma^*$, the overweight of payoff redistribution leads organizations to provide more resources regardless of training overhead and coopetition damage, reducing social welfare. On the other hand, social welfare drops with $\mu$ increases. As depicted in Fig. 11, social welfare decreases as $\mu$ and $\varpi_e$ escalate, suggesting that the impact of intense competition and the training overhead on social welfare cannot be overlooked. *Collectively, the results in Figs. 10 - 11 reveal that an appropriate $\gamma$, e.g., $\gamma^*$, helps maximize social welfare under different competition intensities.*

Fig. 12 illustrates the accuracy of the trained global model and the total data contribution $\sum_{i \in \mathcal{N}} d_i$ under different $\gamma$. According to the definition of TOS, the total data contribution under TOS does not change with $\gamma$ and $\sum_{i \in \mathcal{N}} d_i = 10$. When $\gamma = \gamma^*$, DBR increases the amount of $\sum_{i \in \mathcal{N}} d_i$ by up to 64% with compared to GCA.

**Figs. 13 - 15 depict the training efficiency and accuracy with** $\gamma = \gamma^*$. Figs. 13 - 14 show the loss the trained global model. We fix $|\mathcal{S}_i|$ for each type of the model-dataset in order to fairly compare different baselines. From Figs. 13 - 15, we observe that compared to FIP, WPR and GCA, DBR improves the performance of training efficiency and accuracy. Taking MobileNet-SVHN as an example, DBR improves the accuracy by up to 23.2% compared to GCA. Besides, the results of DBR are close to those of TOS. Although TOS has the best accuracy and speed of convergence, it does not take into account the coopetition damage and training overhead, which limits its application in practical cross-silo FL scenarios.

## VII. Conclusions

In this work, we propose a novel trading mechanism, i.e., TradeFL, to incentivize organizations to provide data and computational resources through mutual trading among organizations. We prove that the interaction among organizations in coopetition is a weighted potential game. Then, we propose a centralized algorithm and a distributed algorithm to determine the optimal resource contribution. TradeFL is conducive to practical adoption as it runs on distributed organizations through smart contracts, providing credibility guarantees for coopetition compensation. Notably, TradeFL does not rely on any specific form of the data-accuracy function. Simulation results demonstrate that our mechanism effectively increases the amount of data contribution and social welfare. Evaluations reveal an interesting observation: increasing the incentive intensity does not always improve social welfare. In the future, we will further consider personalizing the global model assigned to organizations to meet their individual needs.

## Acknowledgment

This work has been partially supported by the National Key R&D Program of China No. 2020YFB1806700, NSFC Grants (No. 61932014, 61972246 and 62232011), Project BE2020026 supported by the Key R&D Program of Jiangsu, China, the Key-Area Research and Development Program of Guangdong Province (No. 2021B0101400003), Hong Kong RGC Research Impact Fund (No. R5060-19), Areas of Excellence Scheme (AoE/E-601/22-R),



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.

General Research Fund (No. 152203/20E, 152244/21E, 152169/22E), and Shenzhen Science and Technology Innovation Commission (JCYJ20200109142008673). Corresponding author: Jie Li and Hongtao Lv. We thank Yuxuan Zhu and Qingshi Zhou's help in improving this paper.

## References

[1] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *IEEE Trans Knowl Data Eng.*, pp. 1–1, 2021, doi:10.1109/TKDE.2021.3124599.

[2] "Swiss re and chinese firm we-bank partner to explore use of ai in reinsurance," https://www.lifeinsuranceinternational.com/news/swiss-re-webank/, accessed Jan. 2022.

[3] "The machine learning ledger orchestration for drug discovery (melloddy) project," https://www.fiercebiotech.com/special-report/melloddy-project, accessed Jan. 2022.

[4] M. Chen, Y. Liu, W. Shen, Y. Shen, P. Tang, and Q. Yang, "Mechanism design for multi-party machine learning," *arXiv preprint arXiv:2001.08996*, 2020. [Online]. Available: https://arxiv.org/abs/2001.08996

[5] H. Lv, Z. Zheng, T. Luo, F. Wu, S. Tang, L. Hua, R. Jia, and C. Lv, "Data-free evaluation of user contributions in federated learning," in *Proc. of WiOPT.*, Oct. 2021, pp. 81–88.

[6] Y. Xue, C. Niu, Z. Zheng, S. Tang, C. Lv, F. Wu, and G. Chen, "Toward understanding the influence of individual clients in federated learning," in *Proc. of AAAI.*, 2021, pp. 10 560–10 567.

[7] N. Ding, Z. Fang, L. Duan, and J. Huang, "Incentive mechanism design for federated learning with multi-dimensional private information," in *Proc. of WiOPT.*, Sept. 2020, pp. 33–40.

[8] N. Ding, Z. Fang, and J. Huang, "Incentive mechanism design for distributed coded machine learning," in *Proc. of IEEE INFOCOM.*, 2021, pp. 1–10.

[9] M. Wu, D. Ye, J. Ding, Y. Guo, R. Yu, and M. Pan, "Incentivizing differentially private federated learning: A multidimensional contract approach," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10 639–10 651, July. 2021.

[10] L. Gao, L. Li, Y. Chen, W. Zheng, C. Xu, and M. Xu, "Fifl: A fair incentive mechanism for federated learning," in *Proc. of ICPP*, 2021, pp. 1–10.

[11] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proc. of AAAI/ACM AIES*, 2020, pp. 393–399.

[12] C. Hasan, "Incentive mechanism design for federated learning: Hedonic game approach," in *Proc. of Springer AAMAS.*, 2021, pp. 1–7.

[13] J. S. Ng, W. Y. B. Lim, Z. Xiong, X. Cao, D. Niyato, C. Leung, and D. I. Kim, "A hierarchical incentive design toward motivating participation in coded federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 359–375, Nov. 2021.

[14] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, Nov. 2020.

[15] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, and H. Shao, "Pain-fl: Personalized privacy-preserving incentive for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3805–3820, Oct. 2021.

[16] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "Fair: Quality-aware federated learning with precise user incentive and model aggregation," in *Proc. of IEEE INFOCOM*, 2021, pp. 1–10.

[17] M. R. Behera, S. Upadhyay, and S. Shetty, "Federated learning using smart contracts on blockchains, based on reward driven approach," 2021. [Online]. Available: https://arxiv.org/abs/2107.10243

[18] Z. Zhang, D. Dong, Y. Ma, Y. Ying, D. Jiang, K. Chen, L. Shou, and G. Chen, "Refiner: a reliable incentive-driven federated learning system powered by blockchain," *in Proc. of VLDB Endow.*, vol. 14, no. 12, pp. 2659–2662, 2021.

[19] L. Feng, Z. Yang, S. Guo, X. Qiu, W. Li, and P. Yu, "Two-layered blockchain architecture for federated learning over mobile edge network," *IEEE Netw.*, vol. 36, no. 1, pp. 45–51, 2021.

[20] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in *Proc. of ACM WWW*, 2021, pp. 947–956.

[21] M. Tang and V. W. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. of IEEE INFOCOM*, 2021, pp. 1–10.

[22] C. Huang, H. Zhang, and X. Liu, "Incentivizing data contribution in cross-silo federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2203.03885

[23] J. Chen, Q. Hu, and H. Jiang, "Social welfare maximization in cross-silo federated learning," in *Proc. of IEEE ICASSP*, 2022, pp. 4258–4262.

[24] S. Kong, Y. Li, and H. Zhou, "Incentivizing federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2205.10951

[25] E. Bonel and E. Rocco, "Coopeting to survive; surviving coopetition," *Int. Stud. Manag. Organ.*, vol. 37, no. 2, pp. 70–96, 2007.

[26] M. Okura and D. CARFI, "Coopetition and game theory." *J. Appl. Econ. Sci.*, vol. 9, no. 3, pp. 458–469, 2014.

[27] X. Wu and H. Yu, "Mars-fl: Enabling competitors to collaborate in federated learning," *IEEE Trans. Big Data.*, pp. 1–11, 2022.

[28] S. Praneeth Karimireddy, W. Guo, and M. I. Jordan, "Mechanisms that incentivize data sharing in federated learning," *arXiv e-prints*, pp. arXiv–2207, 2022.

[29] N. Zhang, Q. Ma, and X. Chen, "Enabling long-term cooperation in cross-silo federated learning: A repeated game perspective," *IEEE Trans. Mob Comput.*, pp. 1–1, 2022, doi:10.1109/TMC.2022.3148263.

[30] F. Chen, J. Wang, C. Jiang, T. Xiang, and Y. Yang, "Blockchain based non-repudiable iot data trading: Simpler, faster, and cheaper," in *Proc. of IEEE INFOCOM.* IEEE, 2022, pp. 1958–1967.

[31] S. Yuan, J. Li, H. Chen, Z. Han, C. Wu, and Y. Zhang, "Jira: Joint incentive design and resource allocation for edge-based real-time video streaming systems," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 5, pp. 2901–2916, May 2023.

[32] A. M. Geoffrion, "Generalized benders decomposition," *J Optim Theory Appl.*, vol. 10, no. 4, pp. 237–260, 1972.

[33] Y. H. Chew, B.-H. Soong *et al.*, *Potential Game Theory.* Springer, 2016.

[34] F. Shi, Z. Qin, D. Wu, and J. McCann, "Mpcstoken: Smart contract enabled fault-tolerant incentivisation for mobile p2p crowd services," in *Proc. IEEE ICDCS*, 2018, pp. 961–971.

[35] S. Yuan, J. Li, and C. Wu, "Jora: Blockchain-based efficient joint computing offloading and resource allocation for edge video streaming systems," *J. Syst. Arch.*, vol. 133, p. 102740, Nov. 2022.

[36] A. Agarwal and Dahleh, "A marketplace for data: An algorithmic solution," in *Proc. of ACM EC*, 2019, pp. 701–726.

[37] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, and Stich, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. of ACM ICML*, July. 2020, pp. 5132–5143.

[38] X. Yao, X. Wang, S.-H. Wang, and Y.-D. Zhang, "A comprehensive survey on convolutional neural network in medical image analysis," *Multimedia Tools and Applications*, pp. 1–45, 2020.

[39] E. Bonel and E. Rocco, "Coopeting to survive; surviving coopetition," *International Studies of Management & Organization*, vol. 37, no. 2, pp. 70–96, 2007.

[40] P. Ritala, "Coopetition strategy - when is it successful? empirical evidence on innovation and market performance," *British Journal of management*, vol. 23, no. 3, pp. 307–324, 2012.

[41] C. I. Fernandes, J. J. Ferreira, P. M. Veiga, and C. Marques, "The effects of coopetition on the innovation activities and firm performance: Some empirical evidence," *Competitiveness Review: An International Business Journal*, 2019.

[42] D. Kwieciński *et al.*, "Measures of competitive intensity–analysis based on literature review," *Central European Management Journal*, vol. 25, no. 1, pp. 53–77, 2017.

[43] X. Zhang, F. Li, Z. Zhang, Q. Li, C. Wang, and J. Wu, "Enabling execution assurance of federated learning at untrusted participants," in *Proc. of IEEE INFOCOM*, 2020, pp. 1877–1886.

[44] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Comput. Appl. Math.*, vol. 124, no. 1-2, pp. 281–302, 2000.



Authorized licensed use limited to: National University of Singapore. Downloaded on November 09,2023 at 12:07:20 UTC from IEEE Xplore. Restrictions apply.