Incentive and Knowledge Distillation Based Federated Learning for Cross-Silo Applications

Beibei Li†, Yaxin Shi†, Yuqing Guo†, Qinglei Kong‡, and Yukun Jiang†
†School of Cyber Science and Engineering, Sichuan University, Chengdu, P.R. China, 610065
‡Institute of Space Science and Applied Technology, Harbin Institute of Technology, Shenzhen, P.R. China, 518000
Email: libeibei@scu.edu.cn; {shiyaxin, yuqingguo, jiangyukun}@stu.scu.edu.cn; kql8904@163.com

***Abstract*—Big data are usually characterized by heterogeneity in real-world cross-silo applications, such as healthcare, finance, and smart cities, leaving federated learning a big challenge. Further, many existing federated learning schemes fail to fully consider the diverse willingness and contributions of data providers in participation. In this paper, to address these challenges, we are motivated to propose an incentive and knowledge distillation based federated learning scheme for cross-silo applications. Specifically, we first develop a new federated learning framework, to support cooperative learning among diverse heterogeneous client models. Second, we devise an incentive mechanism, which not only stimulates workers to provide more high-quality data, but also improves clients' enthusiasm for participating in federated learning. Third, a novel knowledge distillation algorithm is designed to deal with data heterogeneity. Extensive experiments on MNIST/FEMNIST and CIFAR10/100 datasets with both IID and Non-IID settings, demonstrate the high effectiveness of the proposed scheme, compared with state-of-the-art studies.**

***Index Terms*—Artificial intelligence, federated learning, knowledge distillation, incentive mechanism, data privacy.**

I. Introduction

The concept of federated learning was first proposed by Google [1] and was originally aimed at solving the local models' updating problem for Android mobile terminal users. Its architecture requires a global server to orchestrate all training processes, and each client sends the parameters of the local model to the global server for parameter aggregation. During the entire training process, we need to protect local data privacy and prevent some malicious attacks [2]. Under the premise of ensuring data privacy and security, jointing models is implemented to improve the accuracy of the AI models [3].

The efficiency of the conventional federated learning framework depends on the performance of the global server. When the number of clients is very huge, the global server is overloaded and faces a communication bottleneck. In real scenarios, it is unrealistic to design a powerful global server and ensure the server's credibility. Additionally, the goals of federated learning are not uniform, which is a game process. On the one hand, federated learning aims to train a global model for all clients and new participants [4]. On the other hand, clients will sacrifice their individuality in the training process to reach a consensus [5]. During this process, each client has different communication capabilities, different model architectures, and different local data distributions. We regard these conditions as non-independent and identically distributed (Non-IID) [6] characteristics. Compared with independent and identically distributed (IID) data, this heterogeneity of leads to a significant decrease in the local model's accuracy, which can be explained by weight divergence in the model parametric polymerization stage [7]. Therefore, the conventional federated averaging algorithm [1] cannot meet the needs of each client to customize a private customized model.

In the cross-silo scenario, we train data for data islands [8] (e.g., different institutions of the same organization or geographically divided data centers). Moreover, during the application of federated learning in cross-silo scenarios, designing an honest and effective incentive mechanism is also a valuable issue. The goal of the incentive mechanism is to allocate the benefits (e.g., weights, training time, communication bandwidth) generated by the global model to the participating clients, which can encourage data owners to provide more and higher-quality data [9]. A reasonable incentive mechanism can strengthen the security of federated learning architecture's decision-making, improve the efficiency of training process, and combat network attacks [10]. Among them, the incentive mechanism can be reflected in the aggregation of parameters. The result of assigning the same weight under an unbalanced client node load is that the local model performs much better than the global model. By measuring the client's contribution and payoffs, dynamically adjusting parameters is also an effective way to improve the performance of federated learning. The main contributions of this paper are summarized as follows:

- First, we propose a new federated learning framework, to support cooperative learning among diverse heterogeneous client models, achieving the goals of designing local private customized models.
- Second, we devise an incentive mechanism, which stimulates workers to contribute more high-quality private data for local clients and improves the clients' enthusiasm for participation.
- Third, a novel knowledge distillation algorithm is designed to deal with data heterogeneity. Extensive experiments on MNIST/FEMNIST and CIFAR10/100 datasets show the superiority of this algorithm in processing Non-IID data and improving model accuracy.

II. Related Work

In this section, we briefly review the current research progress of federated learning, as well as its combination with knowledge distillation and incentive mechanism.

A. Federated Learning

Conventional federated learning uses the federated averaging algorithm (FedAvg) [1], which was first proposed by McMahan *et al*. Li *et al.* [11] proved that FedAvg is also convergent for Non-IID data. Due to the heterogeneity of data and the inconsistent requirements of clients, the global model is fair to meet the needs of all clients. Therefore, it is inevitable to propose a personalized strategy to optimize each local client. In 2020, Kevin Hsieh *et al.* [12] found that the batch-based normalization method is more likely to fail, and the group-based normalization method has less performance degradation under Non-IID. Based on this, a method of adjusting the communication frequency was proposed, named SkewScout, which can adjust the corresponding communication frequency according to the degree of data's distribution deviation. In 2022, Li *et al.* [13] developed a novel federated learning framework, enabling IoT devices to build comprehensive anomaly detection models in a collaborative and privacy-preserving manner.

B. Incentive Mechanism in Federated Learning

The information asymmetry between model owners and workers, may cause workers to contribute low-quality data or misrepresent data information. As well as different model owners, each profit-maximizing model owner will only maximize their own profits, not the profits of the alliance. In order to solve this problem, Han Yu *et al.* [14] dynamically allocated the given budget to data owners in a context-aware manner. By experiments, they maximized collective benefits through joint training and minimize inequality between data owners. The application of game theory in incentive mechanisms is a hot topic. Shuo Yang *et al.* [15] designed an unsupervised learning method to quantify users, and regarded data quality estimation and monetary incentives as a cooperative game process. Then they used an anomaly detection mechanism to filter abnormal data, improving the quality of the model. In this work, we have improved the work of [16], using incentive mechanism as the measurement threshold of the federated learning framework. Through knowledge distillation, we transfer knowledge between global model and clients, meeting the revenue budget and reaching a consensus.

C. Knowledge Distillation in Federated Learning

In 2006, knowledge distillation [17] was originally designed to extract class probabilities generated by large DNNs or DNN sets, training smaller DNNs with marginal utility loss in 2006. The goal is to deploy complex deep networks in devices with low power consumption and resources while maintaining the accuracy of the model. Then, Jihun Hamm *et al.* [18] combined locally classifiers from different parties to construct an accurate and differentiated private global classifier without leaving the local data. There are various forms of knowledge distillation applications. For instance, Li *et al.* [19] conducted knowledge distillation in federated learning through a shared dataset in 2019, named FedMD, training local private customized models and keeping private data locally. Wang *et al.* [20] use a combination of prompt learning, knowledge distillation, and self-learning to train the DNN neural network. On the basis of satisfying differential privacy, the accuracy and compactness of the Android mobile device models are improved.

III. The Proposed Scheme

In this section, we elaborate on the proposed scheme, first by outlining the structure of the new federated learning framework, then introducing the designed incentive mechanism and the novel knowledge distillation algorithm.

A. The Proposed Federated Learning Framework

The proposed framework designed in this article has a two-tier architecture, as shown in Figure 1. We assume that all participants are selfish and long-sighted. The first layer exists data interaction between local workers and local clients. The local client sends data collection tasks and generates different contract packages $\{R_m, q_m\}$, where R_m represents the payoff that the workers can obtain from the contract and q_m represents the quantity of data that the workers need to provide in accordance with the contract. Workers select the corresponding contract packages to create contracts based on their own type m and upload data in exchange for rewards.

Generally speaking, the local client only has the basic demographic information of the worker. Therefore, the client should collect private sensitive data by constructing a data collection protocol or contract. For example, in medical care, hospitals can collect data such as age, weight, and blood type during outpatient clinics. However, Internet of Things application data of health analysis can only be collected with the consent of the patients.

The second layer of the framework structures a federated learning for different organizations or geographically distributed clients. Different clients may have different model architectures and data distributions (Non-IID settings), but they have to perform the same data classification task. Through knowledge distillation, this data heterogeneity can be turned into advantages. The purpose of training a global general model and designing private customized models can be achieved.

There are N clients in the process of setting up federated learning. Through the first layer, each client has a local labeled dataset $D_k = (x_i^{k \in N_k}, y_i)$, which may come from the same data distribution or different data distributions. In addition, there is a large public dataset D_p, each client can access. After the local collection task is over, each client independently designs its own model f_k to perform the classification task and trains it on the local private dataset to convergence. It should be noted that the model f_k have different model architecture.

After that, each client establishes a collaborative knowledge distillation task to improve the performance of the local

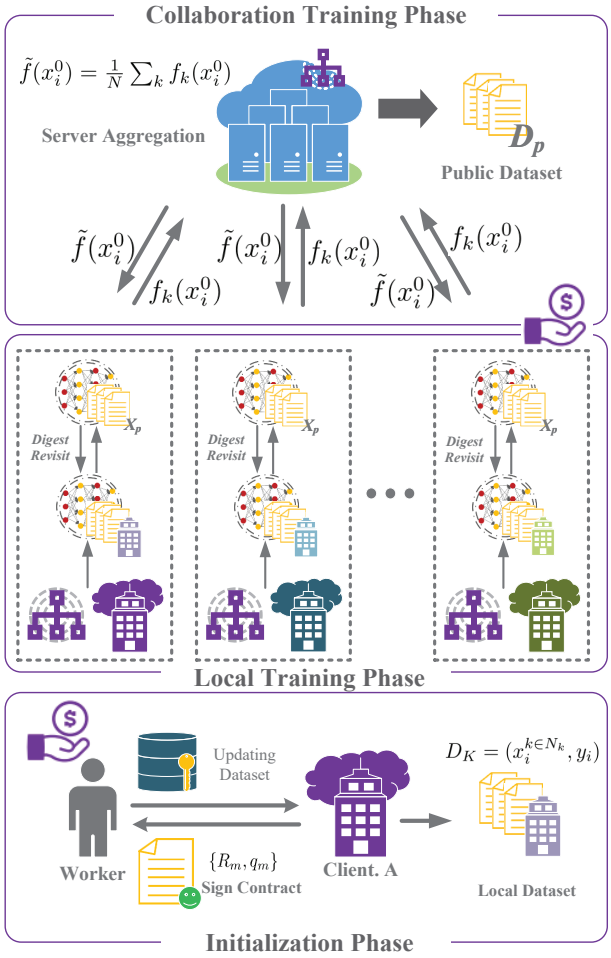

Fig. 1. The proposed federated learning framework.

model f_k, absorbing and integrating the knowledge from other clients.

B. Incentive Mechanism

Algorithm 1 describes the devised incentive mechanism. We focus on a specific client to illustrate the relationship between the worker and the local client, as well as the local models and the global model. The local client issues data collection tasks to workers and generates contract packages $\{R_m, q_m\}$. The payoff that workers can obtain is as follows:

$$Payoff_{(R_m,q_m)}^{u_m} = w_m R_m - c q_m, \tag{1}$$

where w_m is worker's willingness to provide data. Theoretical speaking, the higher the worker's willingness, the higher the quality of the data. c is the cost incurred per unit of data, and q_m is the unit of data. We define that workers of type m will only choose the same type of contract (e.g., $\{R_m, q_m\}$), and will not choose other types of contracts for profit.

The willingness of workers w_m to upload data determines the quality of the data. The quality of the data influences

Algorithm 1: Incentive Mechanism

Initialization phase:
Initializes each client $i \in N$ of type m, contract packages $\{R_m, q_m\}$, workers u_m, and the learning rate μ.
for $i \in N$ **do**
  Each client distributes contract packages $\{R_m, q_m\}$ to workers u_m: $\{R_m, q_m\} \rightarrow u_m$;
  Worker u_m chooses the contract of type m: $\{R_m, q_m\} \leftarrow u_m$;
  Generates local labeled dataset $D_k = (x_i^{k\in N_k}, y_i)$;
end
Local training phase:
Calculate the local payoff $Payoff_i^{Local}$ of the model;
Collaboration phase:
Calculate the federated marginal payoff $Payoff_i^{federated}$;
if Payoff $_i^{federated}$ < Payoff $_i^{Local}$ **then**
  Push unharvested clients out of the federated process: $N = N \setminus \{i\}$;
else
  Refresh the federated cluster N;
end

the time of model training convergence and the number of iterations. The model payoff of the client i can be calculated as follows:

$$Payoff_i^{federated} = Payoff_N - Payoff_{i \notin N}. \tag{2}$$

The federated marginal payoff $Payoff_{federated}$ of the model and its cost determine the value of the local payoff $Payoff_{Local}$. Each client will join the next round of federated learning process, only if the federated payoff $Payoff_{federated}$ is not less than the local payoff $Payoff_{Local}$:

$$Payoff_{federated} \geqq Payoff_{Local}. \tag{3}$$

The clients with negative payoff will be pushed out of the federated learning process and the client cluster will be refreshed. With the iteration of the training, the federated clients are updated to achieve cluster optimization.

C. Knowledge Distillation Algorithm

Algorithm 2 describes the designed knowledge distillation algorithm, which consists of three phases:

- **Step 1:** In the initialization phase, each client issues data collection tasks to workers and generates contract packages $\{R_m, q_m\}$. The user u_m selects contract package of type m, signs an agreement, and uploads data. Each client generates a local private dataset $D_K = (x_i^{k\in N_k}, y_i)$.
- **Step 2:** In the local training phase, each client first trains on the local private dataset D_k to convergence. In order to reduce communication cost and improve training efficiency, the server randomly selects a public subset X_p ($X_p \in D_p$) with a size of 5000. After each client trains

on the public subset X_p to convergence, it sends the class scores $f_k(x_i^0)$ to the server.

- **Step 3:** In the collaboration phase, each client shares the knowledge of their local model by predicting the public subset X_p. The server first averages the class scores $\tilde{f}(x_i^0) = \frac{1}{N}\sum_k f_k(x_i^0)$ uploaded in the local training phase and sends it back to each client. Furthermore, each client trains its model f_k to approach the consensus $\tilde{f}(x_i^0)$.

In typical FedAvg algorithm, the complete dataset is the union of each scattered data, and the loss function is the sum of each private data point's average: $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$. The private data (X_k, Y_k) comes from different distributions $P_k(x, y)$ of k clients. The federated learning on each client can start from the global model copying the weight vector $w^k \in \mathbb{R}^d$. Then, each client performs a local update, and optimizes the local target in several rounds of iterations through the gradient method:

$$\begin{aligned} F_i(w^k) &= \frac{1}{n_k}\sum_{i\in P_k} f_i(w^k), \\ w^k &\leftarrow w^k - \eta\nabla F_k(w^k), \end{aligned} \tag{4}$$

where $F_k(w^k)$ is the loss function of the each client, n_k is the number of local samples, η is the learning rate, and $\nabla F_k(w^k) \in \mathbb{R}^d$ is the gradient vector. It's worth noting that the expectation $E_{p_k}[F_k(w)] = f(w)$ may not be precise, for $P_k \neq P_j$ in the Non-IID settings.

After a period of local update, each client transmits the local model weight w^k to the global server, and then aggregates these weights directly: $w^{global} \leftarrow \sum_{k=1}^{k} \frac{n_k}{n} w^k$, where w^{global} is the weight of the global model, and n is the number of samples of all clients. Repeat the entire training process until the global model is converged.

Being different from the previous federated training, this paper uses the black-box model based on the output class scores of public data samples to transfer knowledge [19]. Among them, the objective function of the server is:

$$\tilde{f}(x_i^0) = \frac{1}{N}\sum_{k}^{N} f_k(x_i^0). \tag{5}$$

Each client aims to $f_k \leftarrow Train(w_i, x_i^{k\in N_k}, y_i)$, so that the local class scores and the output class scores downloaded by the server reach a consensus. Additionally, the weight of each client is worth discussing. When the local data is relatively uniform and the local model of each client is close to the global model, we can use the conventional FedAvg algorithm. On the contrary, when the data distribution is diverse, the FedAvg algorithm is not enough to meet the heterogeneity between clients. We use cosine similarity to measure the weight of each client in the federated learning structure:

$$\tilde{f}(x_i^0) = \sum_{k}^{N} Normal(\cos(w_k, \bar{w}))f_k(x_i^0), \tag{6}$$

Algorithm 2: Knowledge Distillation Algorithm

Local training phase:
Each client trains f_k to convergence on its private D_k:
    $f_k \leftarrow Train(w_i, x_i^{k\in N_k}, y_i)$;
Server randomly samples a public subset $X_p \in D_p$;
Each client trains f_k to convergence on the public subset X_p, and send the class scores $f_k(x_i^0)$ to server:
    $Train(w_i, X_p) \rightarrow f_k(x_i^0) \Rightarrow$ server;
Collaboration phase:
    $\tilde{f}(x_i^0) = \sum_k^N Normal(\cos(w_k, \bar{w}))f_k(x_i^0)$;
for $t \in [R]$ *communication rounds* **do**
    Server randomly samples a public subset $X_p[t+1] \in D_p$;
    for $i \in N$ *clients* **do**
        Each client downloads $\tilde{f}(x_i^0)$ to its local;
        Digest: Each clint $w_i \leftarrow Train(w_i, X_p[t+1])$ to approach the consensus $\tilde{f}(x_i^0)$;
        Revisit: $w_i \leftarrow Train(w_i, x_i^{k\in N_k}, y_i)$ for a few epochs;
    end
end

where $Normal()$ is a normalization function. $\cos(w_k, \bar{w})$ is cosine similarity, measuring the similarity between the client data quality and the global average data quality. w_m is the worker's willingness to provide data and $\bar{w}$ means aggregate data quality. The willingness of workers w_m to upload data determines the quality of the data, affecting the time of model training convergence and the number of iterations.

IV. PERFORMANCE EVALUATION

In this section, we perform experimental analysis on three schemes (the proposed, FedAvg, and FedMD), evaluating the accuracy and incentive effectiveness of the model proposed in this paper when dealing with Non-IID data.

A. Datasets Description and Partitioning

We use two different datasets, named MNIST/FEMNIST and CIFAR10/100, to test the proposed scheme. For MNIST/FEMNIST datasets, the public dataset is MNIST, and the local private dataset is a subset of the FEMNIST. To satisfy the IID setting, the local private dataset is randomly sampled from FEMNIST. To satisfy the Non-IID setting, each client has only one type style of letters by a single writer, but their task is to identify all letters from different writers.

For CIFAR10/100 datasets, the public dataset is CIFAR10, and the local private dataset is a subset of CIFAR100, including 100 subclasses divided into 20 superclasses. Each image has a "fine" label (the class it belongs to) and a "coarse" label (the super class it belongs to). In the IID setting, each client needs to classify the test image into the correct subclasses. In the Non-IID setting, each client only has the image from one subclass per superclass, but their task is to classify each image into the correct superclasses.

TABLE I
SIMULATION EXPERIMENTS COMPARISON TABLE (*Accuracy*).

Settings	The proposed				FedAvg				FedMD			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID	IID	Non-IID
Client 0	0.902	0.897	0.832	0.622	0.812	0.772	0.770	0.543	0.895	0.858	0.835	0.478
Client 1	0.876	0.900	0.780	0.568	0.805	0.771	0.803	0.550	0.886	0.825	0.823	0.583
Client 2	0.814	0.810	0.791	0.504	0.800	0.725	0.780	0.506	0.875	0.867	0.778	0.581
Client 3	0.830	0.826	0.856	0.620	0.816	0.675	0.776	0.567	0.889	0.858	0.818	0.573
Client 4	0.908	0.873	0.828	0.507	0.810	0.781	0.783	0.534	0.885	0.870	0.823	0.584
Client 5	0.875	0.890	0.813	0.653	0.820	0.807	0.793	0.562	0.899	0.901	0.842	0.574
Client 6	0.854	0.851	0.833	0.669	0.823	0.785	0.778	0.544	0.903	0.896	0.845	0.591
Client 7	0.900	0.892	0.794	0.651	0.825	0.727	0.776	0.535	0.903	0.899	0.843	0.568
Client 8	0.917	0.901	0.842	0.643	0.811	0.773	0.751	0.552	0.902	0.900	0.847	0.586
Client 9	0.877	0.883	0.837	0.640	0.795	0.792	0.826	0.546	0.901	0.894	0.807	0.535

B. Baselines Studies

We assume that there are $N = 10$ clients participating in the federated learning process, and the model architecture of each client is two or three-layer deep neural networks. Then, we have compared the following two schemes to show the effectiveness of the proposed scheme.

1) FedAvg [11]: This process defaults that the model quality of all clients is equal, as global aggregation is performed under the condition of equal weight.
2) FedMD [19]: Before the collaboration phase, each participant first trains to convergence on the public dataset. Then they conduct local private data training to reach a consensus with public class scores.

In the incentive and knowledge distillation based federated learning scheme, this process is based on the dynamic adjustment of parameters' weight proposed in this paper. We intentionally reduce the values of the $1th$ and $2th$ clients' willingness w_m and increase $0th$ and $5th$ clients'.

C. Performance Evaluation on Model Accuracy

Figure 2 shows the test accuracy after 10 clients participating in collaborative training as assumed above. We set $R = 20$ and mark a point every two rounds, as Table I. Obviously, all clients can converge to a good performance within 20 rounds, which greatly reduces the communication cost. Compared with the Non-IID setting, the IID setting is more gradual and has faster convergence, due to the complexity of the task. The accuracy of the MNIST/FEMNIST is higher than CIFAR10/100, due to the complexity of the datasets. The accuracy of the model is affected by the willingness of the workers. For example, $1th$ and $2th$ client's model accuracy are slightly lower than others, and the convergence process fluctuates greatly. The curves of $0th$ and $5th$ client models are gentle, and the accuracy is high.

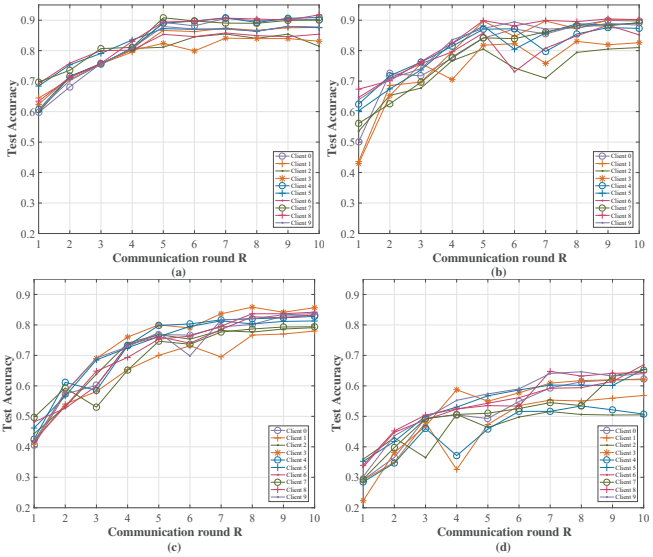

Fig. 2. The model performance of the scheme proposed in this article under the MNIST/FEMNIST and CIFAR10/100 datasets with IID/Non-IID settings. (a) MNIST/FEMNIST IID Setting. (b) MNIST/FEMNIST Non-IID Setting. (c) CIFAR10/100 Non-IID Setting. (d) CIFAR10/100 Non-IID Setting.

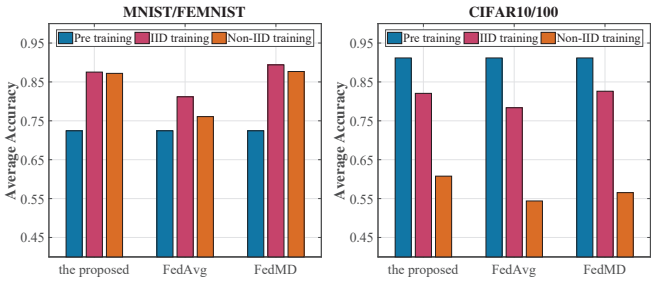

Fig. 3. Comparisons of three schemes (the proposed, FedAvg, and FedMD) which are used to obtain model average accuracy under each setting.

TABLE II
PEARSON CORRELATION COEFFICIENT CALCULATION TABLE.

r_{XY}	MNIST		CIFAR	
	IID	Non-IID	IID	Non-IID
the proposed	0.9397	0.8599	0.7792	0.6134
FedAvg	0.1720	0.4725	-0.2552	0.5882
FedMD	0.3306	0.0945	0.6457	-0.3028

Then, we make comparisons of three schemes (the proposed, FedAvg and FedMD) by recording the average accuracy, under pre-training, IID setting and Non-IID setting. As shown in Figure 3, we demonstrate the superiority of this work in processing complex datasets under Non-IID setting.

D. Performance Evaluation on Incentive Effectiveness

We calculate the Pearson correlation coefficients of various schemes in Table II, which represents the effectiveness of incentive mechanism [16]. When the sample mean exists $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ and $\bar{Y} = \frac{\sum_{i=1}^{n} Y_i}{n}$, the covariance is $Cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{n-1}$. The Pearson correlation coefficient of the sample can be obtained as:

$$r_{XY} = \frac{Cov(X,Y)}{S_X S_Y}, \tag{7}$$

where S_X is the sample standard deviation of X.

In the identical dataset and the data distribution scenario, it can be seen that the proposed scheme has the optimal fairness in distributing client payoffs. The reasonableness of the weights assigned by the federated learning framework is proportional to the Pearson correlation coefficient. This is a quantitative method to measure the contribution of rewarding participants. With the iteration of the federated process, the federated cluster is updated to achieve cluster optimization.

V. CONCLUSION

In this paper, we have proposed an incentive and knowledge distillation based federated learning for cross-silo applications. Significantly, we have measured the payoff of the clients by calculating the incentive relationship between the worker, local model, and federated structure to refresh the federated cluster. We have allowed multiple heterogeneous clients to transfer knowledge in the way of knowledge distillation, creating a local private customized model. Experiments on MNIST/FEMNIST and CIFAR10/100 datasets have proved the effectiveness of the proposed scheme in processing Non-IID data. Moreover, through the calculation of the Pearson correlation coefficient, we have demonstrated the effectiveness of the global server in assigning parameters' weight and optimizing federated clusters. In future works, we will focus on optimizing the incentive mechanism, speeding up the equilibrium convergence, and using more metrics to analyze experiments in diverse perspectives.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, Lauderdale, FL, USA, Apr. 2017, pp. 1273-1282.

[2] B. Balle, G. Barthe, and M. Gaboardi, "Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences," in *Proc. Neural Information Processing Systems (NIPS)*, Montréal, Canada, Dec. 2018, pp. 6280-6290.

[3] R. Shokri and V. Shmatikov, "Privacy-preserving Deep Learning," in *Proc. Computer and Communications Security (CCS)*, Denver, CO, USA, Oct. 2015, pp. 1310-1321.

[4] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated Learning via Over-the-air Computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[5] 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 Transactions on Knowledge and Data Engineering*, 2021. DOI:10.1109/TKDE.2021.3124599.

[6] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and Communication-efficient Federated Learning from Non-IID Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[7] C. Wang, G. Yang, G. Papanastasiou, H. Zhang, and J. J. Rodrigues, "Industrial Cyber-physical Systems-based Cloud IoT Edge for Federated Heterogeneous Distillation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5511–5521, 2020.

[8] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized Cross-silo Federated Learning on Non-IID Data," in *Proc. Artificial Intelligence (AAAI)*, vol.35, no. 9, Feb. 2021, pp. 7865-7873.

[9] D. G. Dobakhshari, P. Naghizadeh, M. Liu, and V. Gupta, "A Reputation-based Contract for Repeated Crowdsensing with costly Verification," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 6092–6104, 2019.

[10] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–physical Systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *Proc. International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, Apr. 2019.

[12] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The Non-IID Data Quagmire of Decentralized Machine Learning," in *Proc. International Conference on Machine Learning (ICML)*, Vienna, Austria, Apr. 2020, pp. 4387-4398.

[13] B. Li, S. Ma, R. Deng, K.-K. R. Choo, and J. Yang, "Federated Anomaly Detection on System Logs for the Internet of Things: A Customizable and Communication-Efficient Approach," *IEEE Transactions on Network and Service Management*, 2022. DOI:10.1109/TNSM.2022.3152620.

[14] H. Yu, Z. Liu, Y. Liu, T. Chen, and M. Cong, "A Fairness-aware Incentive Scheme for Federated Learning," in *Proc. Artificial Intelligence (AAAI)*, New York City, NY, USA. Feb. 2020, pp. 393-399.

[15] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On Designing Data Quality-aware Truth Estimation and Surplus Sharing Method for Mobile Crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.

[16] W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung, and H. V. Poor, "Hierarchical Incentive Mechanism Design for Federated Machine Learning in Mobile Networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9575–9588, 2020.

[17] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model Compression," in *Proc. ACM SIG Knowledge Discovery and Data Mining (SIGKDD)*, Philadelphia, PA, USA, Aug. 2006, pp. 535-541.

[18] J. Hamm, Y. Cao, and M. Belkin, "Learning Privately from Multiparty Data," in *Proc. International Conference on Machine Learning (ICML)*, Anaheim, CA, USA, Jun. 2016, pp. 555-563.

[19] D. Li and J. Wang, "FedMD: Heterogenous Federated Learning via Model Distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[20] J. Wang, W. Bao, L. Sun, X. Zhu, B. Cao, and S. Y. Philip, "Private Model Compression via Knowledge Distillation," in *Proc. Artificial Intelligence (AAAI)*, vol. 33, Honolulu, WV, USA, Jan. 2019, pp. 1190-1197.