Realizing the Heterogeneity: A Self-Organized Federated Learning Framework for IoT

Junjie Pang, Yan Huang, Zhenzhen Xie, Qilong Han, and Zhipeng Cai, *Senior Member, IEEE*

***Abstract*—The ubiquity of devices in Internet of Things (IoT) has opened up a large source for IoT data. Machine learning (ML) models with big IoT data is beneficial to our daily life in monitoring air condition, pollution, climate change, etc. However, centralized conventional ML models rely on all clients' data at a central server, which seriously threatens user privacy. Federated learning (FL) emerges as a promising solution aiming to protect user privacy by enabling model training on a large corpus of decentralized data. The recent studies indicate FL suffers from the heterogeneity issue as it treats all clients' data equally, that is, FL might sacrifice the performance of the majority of clients to accommodate the performance of the minority of clients with low usability data. In order to overcome this issue, a reinforcement learning (RL)-based intelligent central server with the capability of recognizing heterogeneity is implemented, which can help lead the trend toward better performance for majority of clients. To be specific, an FL central server analyses the benefits of different collaboration by capturing the intricate patterns in heterogeneous clients based on rating feedback and then updates clients' weights iteratively, until it establishes a coalition of clients with quasioptimal performance. The experimental results on three real data sets under various heterogeneity levels demonstrate the superior performance of the proposed solution.**

***Index Terms*—Federated learning (FL), heterogeneity, reinforcement learning (RL).**

I. Introduction

THE SIGNIFICANT proliferation of the Internet-of-Things (IoT) techniques and recent rapid advances in machine learning (ML) has resulted in tremendous changes in how people live and work [1]–[3]. More and more users and businesses use IoT devices to process personal, financial, and commercial data, or use them to plan their work and private life, generating a huge amount of personal and business data. The popularity of smart mobile devices (smartphones and tablets) and the booming of IoT applications (apps) in recent years have accelerated this trend, which has necessitated strong needs for novel technologies and devices aiming at intensively sensing the physical world, extensively connecting various things, comprehensively understanding the practical scenarios, and efficiently analyzing the collected data. The investigation of making use of big IoT data through ML algorithms lies at the heart of nowadays technologies. ML-powered IoT [4] thus emerged to help devices to understand the physical world [5], [6] and to make responses by adopting adaptive decision-making [7] methods. Most of the ML models are centralized, that is, the models need to be performed at the edge layer or cloud data center with data collected from different sources or multiple stackholders [8]–[10]. However, clients' privacy is seriously threatened because of the data released to servers, resulting in clients' anxiety, and has become one of the major barriers to the widespread adoption of IoT applications [11].

To tackle the privacy challenges and encourage clients to proactively participate in IoT services, there are continuous efforts to refine every step of intellectual IoT services to optimize the protection of individual privacy. Federated learning (FL) is a collaborative ML method without training data in a centralized manner, by which clients collaboratively learn a common model while preserving personal data on their own devices [12]. Fig. 1 elaborates an edge-based FL implementation. It shows a typical FL protocol utilizing a model updating and sharing mechanism, in which all the clients collaboratively train a global model by providing different data sources and training capabilities. Such heterogeneity refers to clients' different levels of data volume, data quality, and usability, which can significantly affect model training [13].

Traditional FL suffers from the heterogeneity issue as the model assumes different data sources contribute equally to model parameters [14]. When data are not independent and identically distributed (IID) and highly skewed, the performance of collaborative training may be degraded. At the early stage, to resolve the heterogeneity challenge, typical FL aggregation methods treat all clients fairly without an in-depth analysis of the benefits of the mutual relationship among clients. Thus, it is very likely that FL might compromise most clients' accuracy in order to accommodate a small group of clients' performance who has low quality/usability data.

These observations motivate an increasing number of studies to solve the above contradictions between heterogeneity

Manuscript received May 1, 2020; revised June 13, 2020; accepted June 30, 2020. Date of publication July 7, 2020; date of current version February 19, 2021. This work was supported in part by the Program for Innovative Postdoctoral Talents in Shandong Province under Grant 40618030001, and in part by the National Key Research and Development Program of China under Grant 2018YFB2100303. *(Corresponding author: Yan Huang.)*

Junjie Pang is with the Business School and the College of Computer Science and Technology, Qingdao University, Qingdao 266000, China (e-mail: pangjj18@163.com).

Yan Huang is with the College of Computing and Software Engineering, Kennesaw State University, Atlanta, GA 30324 USA (e-mail: yhuang24@kennesaw.com).

Zhenzhen Xie is with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: xiezz14@mails.jlu.edu.cn).

Qilong Han is with the Department of Computer Science Technology, Harbin Engineering University, Harbin 150001, China (e-mail: hanqilong@hrbeu.edu.cn).

Zhipeng Cai is with the Department of Computer Science, Georgia State University, Atlanta, GA 30302 USA (e-mail: zcai@gsu.edu).

Digital Object Identifier 10.1109/JIOT.2020.3007662

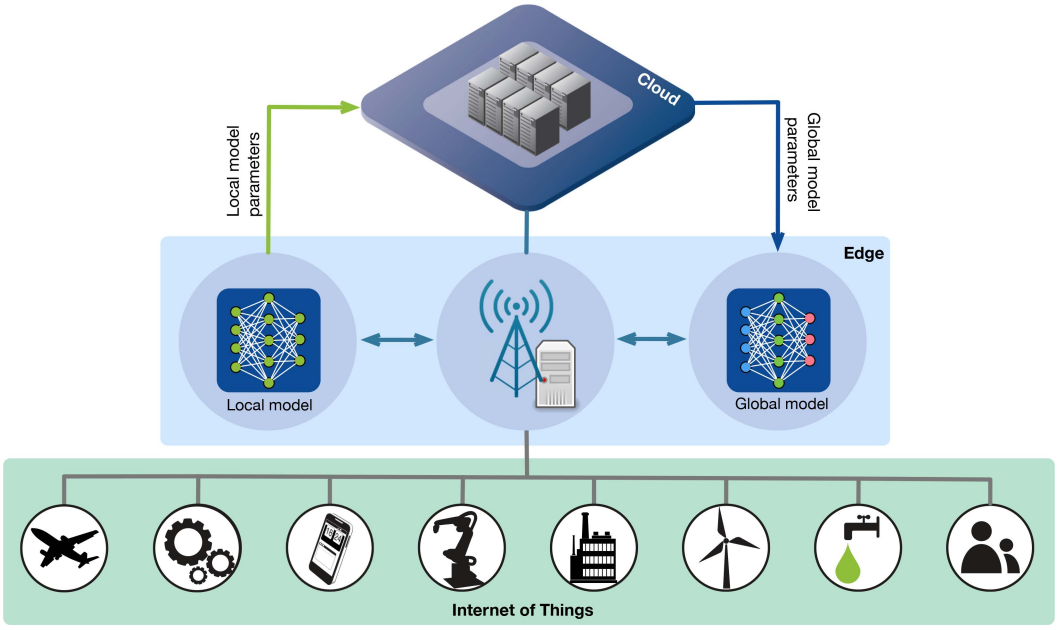


Fig. 1. Overview of edge-based FL for IoT.

of clients, thus improving the performance of clients. The first trend is to minimize the variance of client contributions by keeping individual contribution to be on the same scale [15], [16]. The second trend is to provide fairness guarantee [17] in the aggregation process, like minimizing the variance of client accuracy [18]. However, both of them only try to treat all clients equally since accessing the diversity features like data quality, data quantity, data type can violate the privacy guarantee of FL principles. Fig. 2 illustrates an example of the issue. We assume Clients 1–4 have high-quality data and behave normally, where Client 4's data type and data distribution are different from others'. Client 5 is a data poison attacker or has low-quality data. We expect FL provides high performance for Clients 1–3 as they represent data from the majority clients. With the current works, the accuracy of Clients 1–3 will be sacrificed to compromise Clients 4 and 5 for fairness. More problematically, the performance of Clients 1–4 will be dramatically decreased because Client 5 launches a data poisoning attack.

Under such circumstances, an FL central server in the current solutions cannot recognize that some clients' data cannot be effectively utilized to collaborate with other clients. Therefore, FL needs an exploration capability to obtain a satisfying collaboration plan for clients to deal with heterogeneity rather than simply minimizing variance.

To fill the gap, we propose a solution for an FL central server to have the cognitive capability of recognizing heterogeneity in order to generate a collaboration plan with a high-performance increment for clients with beneficial relationships, instead of offering clients a "fair" allocation principle by compromising training performance. Specifically, the collaboration plan refers to a quasioptimal weight setting on clients' contributions in our work. To achieve the goals, a self-organized evolutionary process with client feedback is introduced to obtain a suitable collaboration plan. This process is powered by a reinforcement learning (RL)-based collaboration plan searching algorithm, which employs an adaptive weighting mechanism based on client feedback. Our solution offers two functions: 1) every client can provide feedback in the current collaboration during the training process and 2) the FL central server can lead the trend to accommodate most clients' data and distinguish the ones that fail to cooperate.

By recognizing heterogeneity rather than actions to achieve certain predefined optimization goals, FL can better satisfy most client expectations on model performance. We illustrate the proposed design in Fig. 2. Through this RL-based self-organized evolutionary process, Clients 1–3 will finally form a stable collaboration plan, since they all have similar feedback in most training iterations. Meanwhile, Clients 4 and 5 will be inhibited because their performance goal can only be satisfied with the expense of others. It is noted that if Client 5 happens to be a data poisoning attacker, the effects will be minimized since Client 5 may have a negative impact. To successfully form the collaboration plan, we use the Markov decision process (MDP) to formalize the self-organizing concept. An agent observes clients' feedback from the last collaboration in terms of weight settings (states), converts the feedback into a numerical reward value (reward), and adapts every client's weight in the direction (action) that would result in better feedback. During an RL training process, each iteration is an attempt to a new and better collaboration that needs further evaluation. To not disturb privacy settings in FL, we only add the rating feedback in terms of local model accuracy to provide an individual evaluation of the current weight setting. When we finally obtain an stable collaboration plan, the clients who are incompatible with others will be compromised, in which case they have a chance to leave to join another group rather than waiting for a model with unsatisfactory performance. Eventually, the proposed framework can track the empirically optimal weight

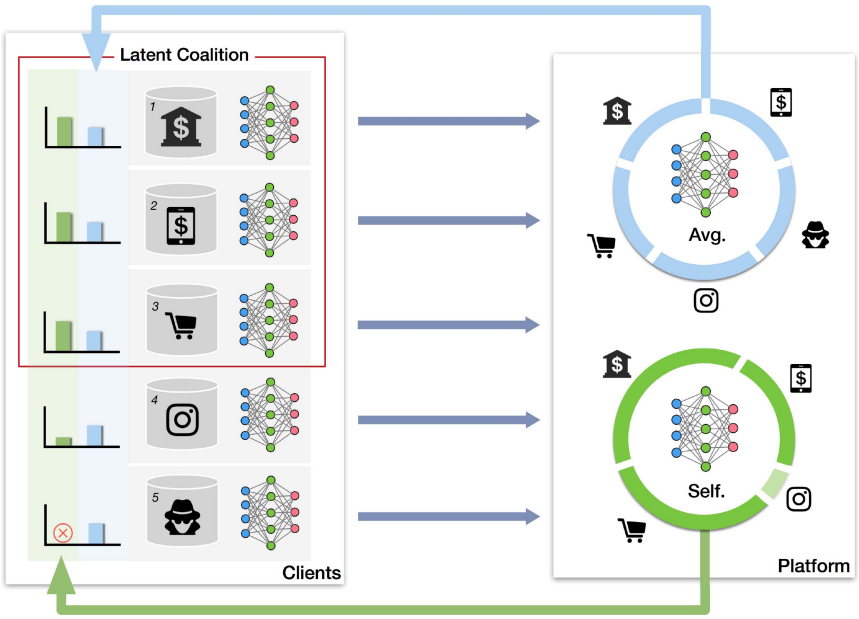


Fig. 2. Client heterogeneity in FL.

settings to help most clients achieve high performance consistently. Our contributions can be summarized as follows.

1) *Realizing Heterogeneity ia Self-Organizing Concept:* The proposed aggregation method can automatically create different collaboration schemes to identify the heterogeneity hidden in the federation. It utilizes an active search to find a stable collaboration plan for better performance.
2) *RL-Based Optimized Aggregation:* A deep deterministic policy gradient (DDPG) [19]-based aggregation method called C-DDPG is proposed to update and manage the collaboration plan automatically.
3) *Inherent Quality of Blocking Data Poisoning Attack:* The proposed collaboration plan searching process can prevent data poisoning attacks.

The remainder of this article is organized as follows. Section II introduces related works. The detailed structure of self-organized FL is presented in Section III. Section IV explains the implementation details. The experiments and results are analyzed in Section V. Finally, conclusion and future work are presented in Section VI.

II. Related Works

Recently, many learning methods emerge for intelligent IoT applications in various scenarios, such as the tie direction learning approach called DeepDirect in mixed social networks [20], fragile recognition image (FRI) recognition for environmental condition analysis [21] and transfer learning-based object detection like a low-shot transfer detector framework called LSTD [22]. Besides the benefits provided by ML-powered IoT applications, there are several challenges lying in multimodel data analysis and privacy protection [23]. Distributed ML and FL are regarded as promising solutions to tackle those challenges. In this section, a brief review of several recent relevant works is presented.

Distributed Learning and FL: The increasing amount of IoT data and the higher precision requirements necessitate the implementation of distributed learning techniques to resolve the concerns on computation efficiency and data analysis. First, a distributed ML [24] approach is proposed with the intention of dealing with massive data scale and gigantic model size. After that, a number of improved frameworks with better convergence rate [25], [26] have been proposed. Those models also have better capability to work on multi-model data [27]. Distributed learning has been proven to be effective and efficient to reduce the learning errors on large complex data. However, all the client data are required to be released to servers, resulting in serious privacy and security issues [28]–[30]. In order to overcome the privacy and security concerns of clients, Google proposed the FL framework, aiming at enabling model training on a large corpus of decentralized data [29], [30]. Unlike distributed ML, the central server in FL only needs to maintain and update the clients' training parameters in an iterative manner without manipulating raw data from each client. Besides taking into account of privacy issues, FL also can make use of individual computation resource to accomplish complex training tasks among distributed devices.

Heterogeneity Problems in FL: Various clients with heterogeneous data (e.g., data type, data quantity, data quality, etc.), diverse computation capabilities, and different service expectations are involved in FL. Heterogeneous data can promote mutually beneficial resources for FL when data are complimentary to each other. On the other hand, heterogeneity also brings unprecedented challenges as low quality, low utility, or malicious data might jeopardize learning performance. FL may sacrifice performance in order to accommodate those unsuitable data.

Fairness mechanisms have been proposed recently to tackle the heterogeneity issues. Such mechanisms try to balance the impact of individual client in terms of contribution, accuracy, etc., on the global model such that no client data can dominate the model training procedure. The widely adopted fairness goals include minimum accuracy [31], equal error rate [17], and min/max fairness [32]. Most works aim at finding a tradeoff among fair allocation and individual expectation. To name some, the work in [31] uses a fair allocation solution to obtain fairness for clients. The work in [33] optimizes the worst performance of an individual device under a rate constraint. However, minimizing the variance by setting a fairness constraint could result in that many major clients' performance could be sacrificed to satisfy the fairness goal. Instead of setting a unified fairness index for an individual client, the work in [18] proposes to protect the fairness of a group of clients with similar data. To implement the idea, a data-dependent Rademacher complexity guarantee is proposed to solve the data heterogeneity problem and it reduces the bias on the multidomain data set during the training process. In their FL model, FL performance highly relies on clustering results. However, determining the similarity of clients requires additional knowledge which might be impractical for some IoT applications. Furthermore, additional knowledge could also cause privacy issues which contradict with the original intention of FL design.

Another vein of research targeting at the heterogeneity issues is to design incentive mechanisms in FL to minimize the behavior variance among clients to improve the performance of the global model [34], [35]. For most FL incentive mechanisms, clients are motivated by rewards to preserve their contribution quality. For example, the work in [36] implements this idea by using a contract mechanism to translate the contributed resources to appropriate rewards. In general, FL with an incentive mechanism requires effective reward settings and a precise evaluation to measure the impacts of clients. Unfortunately, incentive mechanisms cannot solve heterogeneity issues when there are low quality, low utility, and even malicious data due to the lack of consideration about their impact.

To tackle the heterogeneity challenges, we balance the global model performance and data diversity by a novel self-organized process with RL support. Instead of using a unified fairness constraint, our proposed method can adaptively generate a collaboration plan for preserving good performance for most clients, while reducing the negative impact of low quality, low utility, or malicious data.

III. Self-Organized FL Aggregation Framework Design

In this section, we first present the background of the FL aggregation process and introduce the typical implementation of FedAvg. After that, we elaborate on the heterogeneity issues in FL and explain the necessity for designing the adaptive framework in order to alleviate the negative impact of heterogeneity issues. Finally, the details of our adaptive framework are introduced.

A. Preliminary

Compared with distributed learning, FL adopts model sharing instead of data sharing. A typical FL model utilizes a novel collaborative training platform that maintains clients' private data locally while achieving a high-performance global model. The training process can be described as follows.

Primary Initialization (Phase 1): For a new FL task T, the central server S trains an initial global model on public data set.

Local Training (Phase 2): The initialized global model is fed to local clients and trained on each client's data set to update a local parameter.

Local Parameter Aggregation (Phase 3): The parameters from local clients are aggregated in order to improve the global model.

Iteration (Phase 4): The distributed update and the centralized parameter aggregation are repeated for a certain number of times until the global model converges or achieves an expected performance goal.

FedAvg [12], one of the widely adopted aggregation algorithms in FL is introduced. Assume there are k clients and the kth client holds a training data set $x_{k,1}, x_{k,2}, \ldots, x_{k,m_k}$. The objective of FedAvg is formulated as an optimization model

$$\min_{\mathbf{w}} \left\{ F(\mathbf{w}) \triangleq \sum_{k=1}^{N} p_k F_k(\mathbf{w}) \right\}$$

where $p_k = (m_k/m)$ and the local objective $F_k(\cdot)$ is defined by

$$F_k(\mathbf{w}) \triangleq \frac{1}{m_k} \sum_{j=1}^{m_k} f\left(\mathbf{w}; x_{k,j}\right).$$

FedAvg performs the aggregation by averaging the parameters from individual local models. Note that, each client's behaviors could be different in terms of data volume, training capability, the expectation on performance improvements, and data distribution, the heterogeneity problem thus becomes a critical factor that affects FL aggregation result. Aiming at this challenge, FedAvg can only solve the volume variance among clients by allocating large weights to the clients with a larger data set, which inevitably sacrifice the other clients who have less data volume but high data quality. Differently, other aggregation methods tend to treat each client equally to alleviate the impacts of heterogeneity: they minimize the accuracy gap between each client's local model to ensure clients with less accuracy can receive more attention. However, they fail to identify different individual client's expectation and correlations among clients' data distributions, which are also the primary reasons can lead to heterogeneity. Thus, we reconsider the heterogeneity issue by taking the above two factors into the aggregation process. Furthermore, we expect such an aggregation process can be implemented by adding minimal changes to several primary principles in FL, like privacy guarantee and collaborative training feature.

B. Deep Insight Into Heterogeneity of FL

To fully explain our perspective on resolving the heterogeneity issue, we investigate the mutual beneficial correlations

and mutual inhibited correlations among client data during collaborative training in FL.

Assume a collaborative learning model has several initial clients *client* 1, *client* 2, ..., *client* 7. Intuitively, when there is a close correlation between *client* 1, *client* 2, ..., *client* 5, it indicates that all of them will mutually beneficial to each other during the collaborative training process. Under such conditions, the global model can have significant improvements and each of the clients will receive a better local model. When *client* 6 and *client* 7 are the dissimilar client who has less correlation between *client* 1, *client* 2, ..., *client* 5, FedAvg liked aggregation method will generate a new collaboration plan by allocating different weight with them, which apparently could impact the benefits of *client* 1, *client* 2, ..., *client* 5. If *client* 7 is with a high data volume, it can aggravate this situation. Under such conditions, since the FL central server has insufficient knowledge of the correlations among these three clients, *client* 7 will have the largest weight since his contribution seems to be critical. That means, in the worst case, *client* 1, *client* 2, ..., *client* 6 may finally receive a global model that has fewer performance improvements than the previous collaborative training result.

The aforementioned observations motivate us to collect knowledge of correlation among clients with explicit evidence. Thus, we decide to design an intelligent central server that can automatically generate multiple collaboration plans and compare their performance improvements, until there is an optimal one that can satisfy most of the client's expectations of local model performance. That means, when *client* 1, *client* 2, ..., *client* 5 have formed a solid collaboration, and FL central server finds that *client* 6 and *client* 7 cannot collaborate with either of them by attempting multiple weight allocation forms in different collaboration plan, we tend to allocate less weight to hold *client* 6 and *client* 7's training updates for protecting the performance gains of *client* 1, *client* 2, ..., *client* 5.

To ensure this aggregation process will not disturb the privacy guarantee of FL, we add a feedback mechanism in terms of rating to help the central server deduce the correlation among clients. Then, the FL aggregation process is improved as a collaboration plan searching process: the FL central server model iteratively searches the better collaboration plan based on each client's feedback, until it becomes stable. In each collaboration plan, the weight allocation adjustment toward protecting the clients with larger performance improvement. Finally, a stable collaboration plan can be achieved, in which every client can receive a model through the best collaboration. Meanwhile, the other clients outside this plan is paid less consideration since our method expects to protect the clients that can collaborate well, rather than sacrificing several clients' performance that could have larger improvements. For clarity, we give the formalization as follows.

Given $C = \{c_1, c_2, \ldots, c_n\}$ as n clients in the FL platform,and we define g_{avg} to represent a baseline performance index, which can be calculated as follows:

$$g_{\text{avg}} = \frac{1}{|C|} \sum_{c_i \in C} g_{\text{local}}(c_i)$$

where $g_{\text{local}}(c_i)$ is the performance gain when client c_i train the model by the local raw data. Then, we use a weight allocation to denote a collaboration plan p_i, where $p_i = \{w_i^1, w_i^2, \ldots, w_i^n\}$. Under the collaboration plan p_i, we have the performance gain of each client c_i to be denoted as $g_{p_i}(c_i)$. Thus, a coalition u_{p_i} is denoted as

$$\begin{cases} c_i \in u_{p_i}, & \text{when } g_{p_i}(c_i) > g_{\text{avg}} \\ \text{others}, & \text{when } g_{p_i}(c_i) \leq g_{\text{avg}}. \end{cases}$$

Note that, a coalition indicates that each member can receive a positive performance improvements by such a collaborative training since each client can achieve higher $g_{p_i}(c_i)$ than g_{avg}. Thus, we can calculate the performance gain $G(p_i)$ for collaboration plan p_i by the following:

$$G(p_i) = \frac{1}{|u_{p_i}|} \sum_{c_j \in u_{p_i}} g_{p_i}(c_j).$$

Thus, our objective is to find a collaboration plan from $P = \{p_1, p_2, \ldots, p_m\}$ as the optimal weight allocation setting, which can represented as

$$p_* = \arg\max_{p_i \in P} G(p_i)$$

where the coalition in p_* can receive the optimal performance improvements.

C. Novel Collaboration Plan Discovering Method to Solve Heterogeneity Problem in FL

Before diving into the implementation details, we introduce our proposed FL aggregation process based on the self-organizing concept first. Afterward, we give the detailed working process about how to search for a collaboration plan that represented by an optimal weight allocation setting p_* that tends to protect the clients inside a coalition can receive the maximized performance improvements.

Different from existing aggregation methods, we regard the FL aggregation process as a self-organizing system [37], in which each client c_i can recognize how the collaboration impacts their income and the collaboration plan p_i can always toward better performance achievements $G(p_i)$. To achieve such a self-organizing-based FL aggregation process, we introduce a feedback mechanism to collect every client's evaluation instead of the direct interactions among clients. This feedback mechanism can effectively reflects how the collaboration plan p_i satisfy the client's expectation on $g_{p_i}(c_i)$ without disturbing the privacy settings of FL principles. To be specific, we assume the clients submit their feedback to the central server to rate the current collaboration plan p_i in each iteration. After that, the updated collaboration plan p_{i+1} based on the new rating feedback will help the central server adjust the current collaboration plan. The process is repeated until the performance improvements G becomes stable. In this process, the rating feedback refers to individual performance changes of the current collaboration plan over the previous one. Algorithm 1 further depicts the details, and our RL-based automatic collaboration plan searching method is applied to automatically find the p_*.

Algorithm 1 Self-Organizing-Based FL Aggregation

Input: Client Local Training Model
Output: Optimized Collaboration Plan

1: *FL_Initialization*()
2: *Waiting_Clients*() // Will be activated at any moment if a new client joined.
3: **if** Initialized OR New Client Joined **then**
4: *All_Client.Local_Training*(*Period T*)
5: *All_Client.upload*(*Local_Parameter L*)
6: $W = FL.Client.rating()$
7: $G = FL.Aggregation(W)$
8: *All_Client.Stay_Or_Leave*(*W*)
9: **repeat**
10: *All_Client.download*(*G*)
11: $L_{aggre.} = All\_Client.aggregation(G, L)$
12: $L_{+1} = All\_Client.train(L_{aggre.})$
13: *Client.upload*(L_{+1})
14: **until** Converge
15: **end if**

In the initialized step (line 1), the FL platform is open to each client who have a similar training goal. Furthermore, the training task keep open throughout the aggregation process (line 2), which means new clients can join the training and find their location in the collaboration plan with no opening-time constraint. When each client accomplishes the local training and upload the local parameter to the FL central server, it triggers the aggregation process. As an initial setting for the proposed weight allocation, each client is assigned equal weights at first. Afterward, each client upload the local parameter to the FL central server and wait for the aggregation result (lines 4 and 5). Next (line 6), each client gives the rating feedback to the FL central server. According to the consensus lies in current feedbacks, FL central server makes the reweight decision (line 7). Finally, a stable collaboration plan is achieved when the result converges or after a certain number of iterations (lines 10–13). Those clients that eventually cannot collaborate with the majority can decide to leave for this group and join another one (line 8).

IV. Implementation: Self-Organized Process With RL Support

In this section, the RL technique is introduced to implement the self-organizing-based FL aggregation process.

RL is a learning goal-directed learning approach; it uses an interactive manner to explore the environment and investigate how an agent can derive the maximum accumulated reward. As typical settings, *states* include the primary information of the environment. The agent learns how to make *actions* at each state, and finally find the optimal action-state mappings (optimal policy) to maximize the cumulative *reward* as a learning result. By such a dedicated abstraction for decision making under an environment full of uncertainty, it can explore the environment automatically and recognize the optimal policy to help the agent always make right decisions under different conditions. From the above facts, we observe an explicit relation between RL techniques and our self-organizing process, which can be fully explained by the following MDP formulation.

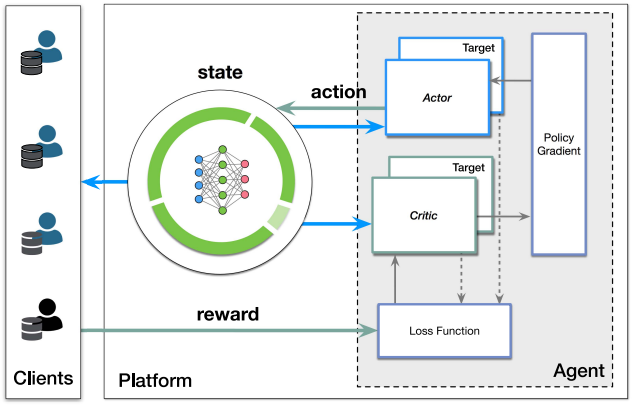


Fig. 3. RL-powered coalition discovering.

In our proposed framework, the FL central server can be regarded as an RL agent, it actively collects clients' feedbacks to optimize the weights allocation, thus enhancing the performance increment of the clients in the optimal coalition. The environment is the object that the RL agent interacts with, which includes the FL central server, clients feedback, and performance improvements of both local model and global model. The state in our MDP is the current collaboration plan p_i. The action determines the changes of current weight allocation. The reward is represented by the performance gap between clients. This MDP guides the RL agent to gradually identify an optimal policy and find the optimal p_* by maximizing the accumulating rewards.

A. RL Model Formalization

To implement the proposed aggregation method with RL support, we introduce the detailed formulation of the MDP process, which represent our design as a sequential decision-making problem, and describe how we model the state, action, and reward. The whole RL diagram is depicted in Fig. 3, and the formation are denoted as follows.

States Reflect the Current Collaboration Plan: We define the FL central server as the RL agent in the self-organized aggregation process and it observes the current global model's performance, weight settings on each client's contribution and feedback on the global model. The goal of the RL model is to find a weight allocation to form a collaboration plan p_i that can satisfy majority clients and make sure their performance increment will be enhanced. Thus, in our MDP, the state S is defined as $S = \{s_1, s_2, \ldots, s_t\}$, where state s_k is the kth rating epoch. Note that, the state definition is the same with collaboration plan, which means $s_k = p_k$.

Actions Are the Changes on Each Client's Weight: We use a set of weight changes to state s_i to represent an action. Each action can generate a new collaboration plan. Thus, we define an action a_k at state s_k as $a_k = \{a_1^k, a_2^k, \ldots, a_n^k\}$. For the whole action set $A = \{a_1, a_2, \ldots, a_j\}$, it includes all the possible collaboration plan changes. Throughout the MDP, to find a

Algorithm 2 Coalition Searching Based on DDPG (C-DDPG)

Input: actor network $\pi(s|\theta^{\pi})$, target actor network $\pi'(s|\theta^{\pi'})$, critic network $Q(s, a|\theta^{Q})$, target critic network $Q'(s, a|\theta^{Q'})$
Output: Optimal actor network and critic network weights θ^{π} and θ^{Q}

1: Initialize network weights θ^{π}, θ^{Q}, $\theta^{\pi'} \leftarrow \theta^{\pi}$, $\theta^{Q'} \leftarrow \theta^{Q}$
2: Initialize replay buffer R
3: **while** In each iteration **do**
4: Choose action $a_t = \pi(s_t|\theta^{\pi}) + noise$, receive client's feedback and calculate r_t, observe new state s_{t+1}
5: Store transition $\{s_t, a_t, r_t, s_{t+1}\}$ in R
6: Sample N examples $\{s_n, a_n, r_n, s_{n+1}\}$ from R
7: Compute the target value y_n and update critic Q to minimize the loss L
8: Update actor π using the sampled policy gradient $\nabla_{\theta^{\pi}} J$
9: Update target networks:
$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$
$\theta^{\pi'} \leftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$
10: **end while**

better coalition, the RL agent changes the collaboration plan by selecting various actions that guided by a reward function.

Reward Is Bounded to the Self-Organized Goal Under Specific Consensus: Before the agent takes actions to reweight each client to form a new collaboration plan p_{i+1}, it obtains every client's feedback based on the last collaboration plan p_i. Here, the accuracy from a local client is adopted as his feedback, and we use it as reward for our MDP to guide the RL process. The reward function indicating performance improvement is calculated as follows: first, g_{avg} represents the average accuracy when the client trains the model separately. Then, at state s_k, we use $g_{s_k(c_i)}$ to represent the accuracy of client c_i. Then, we can divide the clients into two categories: $c_i \in C^k$ if $g_{s_k(c_i)} \geq g_{\text{avg}}$, and otherwise $c_i \in U^k$. Therefore, the reward function is set as follows:

$$r_k = g_c^k - g_u^k, \quad \text{where} \quad \begin{cases} g_c^k = \frac{\sum_{c_i \in C^k} g_{s_k(c_i)}}{|C^k|} \\ g_u^k = \frac{\sum_{c_i \in U^k} g_{s_k(c_i)}}{|U^k|}. \end{cases}$$

As a result, the maximum reward is the significant improvement of a collaboration plan that over the previous collaboration plan. Meanwhile, the impact of the clients outside the current coalition is reduced by dropping their weights. Note that the clients outside the final collaboration plan always represent those clients who have data with low usability to the inner-collaboration clients.

B. Coalition Searching Based on DDPG

We propose a coalition searching method called C-DDPG that implements a novel aggregation process based on an advanced deep RL algorithm, DDPG, as it can provide an accurate representation of the collaboration plan changes (Algorithm 2).

In C-DDPG, we need to train an actor network and a critic network as shown in Fig. 3. The actor network (regard as the environment exploration) is used to generate different actions. Meanwhile, the critic network will evaluate the actor network to make sure it will be updated in the direction toward maximum accumulating rewards.

Actor Network: To interact with the environment, we use the actor network to generate different action a_t by using policy function $\pi(s|\theta^{\pi})$

$$a_t = \pi\left(s_t|\theta^{\pi}\right) + \text{noise}$$

where the noise is used to ensure the exploration through training. In this article, the policy function is estimated by a deep neural network (DNN) with three fully connected layers, which takes the current state as input and then output an action.

To ensure the policy will be improved toward higher accumulating rewards, the parameter θ^{π} in the policy function is be updated using gradient $\nabla_{\theta^{\pi}} J$ (line 8)

$$\nabla_{\theta^{\pi}} J = \frac{1}{N} \sum_n \Big[\nabla_a Q\Big(s_n, \pi(s_n)|\theta^{Q}\Big) \nabla_{\theta^{\pi}} \pi\left(s_n|\theta^{\pi}\right) \Big]$$

where $Q(s, a|\theta^{Q})$ is an action-value function.

Critic Network: To ensure the actor network can be updated toward right direction, DDPG uses the critic network to evaluate actor network's policy. The critic network will estimate a Q value of each (s_t, a_t). The $Q(s_t, a_t)$ indicates the accumulating reward that weight adjustment action will receive under a collaboration plan, which can be calculated by the Bellman optimality equation as follows:

$$Q\Big(s_t, a_t|\theta^{Q}\Big) = \mathbb{E}\Big[r_t + \gamma Q\Big(s_{t+1}, \pi(s_{t+1})|\theta^{Q}\Big) \Big]$$

where the parameter θ^{Q} is updated by minimizing the loss function L, using a mini-batch experience

$$L = \frac{1}{N} \sum_n \Big(y_n - Q\Big(s_n, a_n|\theta^{Q}\Big) \Big)^2$$

in which y_n is the target value and can be obtained by

$$y_n = r_n + \gamma Q'\Big(s_{n+1}, \pi'\Big(s_{n+1}|\theta^{\pi'}\Big)|\theta^{Q'}\Big)$$

where $\gamma \in [0, 1]$ is the discount factor, Here, Q' and π' are the target networks that we will introduce them latter. The gradient of L is calculated as

$$\nabla_{\theta^{Q}} L = \frac{1}{N} \sum_n \Big[2\Big(y_n - Q\Big(s_n, a_n|\theta^{Q}\Big) \Big) \nabla_{\theta^{Q}} Q(s_n, a_n) \Big].$$

During the critic network training, each of the selected action will be evaluated by the primary critic network (line 7). Thus, so that the actor network can be updated toward the critic network suggested direction. At the same time, the critic network is also be updated according to the reward to make sure the suggested direction is leading to maximum accumulating rewards.

Target Network and Experience Replay: In our proposed algorithm, the target networks are two copies of their original actor network and critic network (line 9). They aim to constrain the target values to change slowly so that both Q function value and policy function would not be over-estimated.

Here, the "soft update" can be calculated as follows:

$$\theta^{Q'} = \tau\theta^{Q} + (1-\tau)\theta^{Q'}$$
$$\theta^{\pi'} = \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$$

where $\tau \in [0, 1]$.

To further enhance the training efficiency, we use a mini-batch experience $\{s_n, a_n, r_n, s_{n+1}\}$, $n \in \{1, \ldots, N\}$ at each training step, which is randomly sampled from the replay buffer. To have a better balance between exploitation and exploration, at each training step, our algorithm utilizes the replay buffer technique to ensure the training data to be independently distributed, which refers to a mini-batch experience $\{s_n, a_n, r_n, s_{n+1}\}$, $n \in \{1, \ldots, N\}$ sampling from replay buffer of a fixed size.

V. Simulation

This section validates our proposed self-organizing FL aggregation method through extensive experiments. First, we give the applied data sets and critical experiment settings of RL. Then, multiple aspects of experiment results, comparisons, and analysis are provided, including the comparisons between FedAvg and our proposed method in terms of accuracy improvement comparisons, the performance gap analysis between well-collaborated clients and others, and the effectiveness of catching the strong correlations among clients.

A. Data Sets and Experimental Settings

Data Sets: We evaluate the performance and robustness of our trained model using three data sets, including MNIST, Fashion MNIST, and CIFAR10.

For each data set, there are 1000 clients assigned to different categories. The assigned data is the private data of each client in the FL scenario. Clients have no knowledge about other clients' categories. To represent the heterogeneity feature in the proposed scenario, we assume 500 clients (user ID from 0 to 499) have their data sets of the same categorical distribution to form a simulated coalition. Another 450 (user ID from 500 to 949) clients have a random distribution data assignment with a significant difference. The rest 50 (user ID from 950 to 999) clients are the adversaries with fabricated data to model the data poisoning attacks. With the above simulation setting, we compare our proposed C-DDPG with FedAvg from the aspects of accuracy and efficiency. Furthermore, we validate that the coalition is formed in the simulation.

Experiments Settings and Parameters: We implement our model on the PyTorch platform. For the proposed RL-based self-organized algorithm (C-DDPG), both the actor and critic neural networks are DNN-based. The replay buffer size is 10 000, and the mini-batch size for sampling is 32. We set all the learning rates of the actor and critic neural networks as 10^{-4}, discounting factor γ as 0.99, and learning step size τ as 10^{-3}.

Baseline Algorithm: Since no previous works have studied the FL aggregation process with feedback mechanism or resolving the heterogeneity problem considering both correlation analysis and clients' expectations, we select FedAvg as our baseline algorithm.

TABLE I
Statistics of the Test Accuracy Distribution for C-DDPG

Dataset	**Objective**	FedAvg (Average)	C-DDPG (Average)
MNIST	inner-coalition	85.23%	**87.93%**
	others	21.90%	**12.24%**
Fashion_MNIST	inner-coalition	90.72%	**92.15%**
	others	27.92%	**22.45%**
CIFAR10	inner-coalition	66.05%	**68.56%**
	others	0.91%	**0.15%**

Evaluation Metrics: To evaluate our proposed aggregation method, three evaluation metrics are adopted in our simulations: first, model accuracy in three different application scenarios are given to represent the performance achievements; second, model accuracy in different heterogeneity level describes different aggregation method can affect the clients' performance achievements significantly; and third, the captured correlation among clients in various kinds of scenarios indicates if our proposed method can successfully achieve the quantitative characteristic of positive collaboration.

B. Performance Comparison via Multiple Scenarios

We first compare the accuracies of C-DDPG and FedAvg by using three different scenarios. The results in Fig. 4 present the accuracy performance of C-DDPG and FedAvg for MNIST, Fashion MNIST, and CIFAR10. The *x*-axis represents user ID, and the *y*-axis represents accuracy. The results indicate trading off fairness leads to better performance on inner-coalition clients. Both of the two methods provide meaningful results for the clients in the stable collaboration plan. We observed that C-DDPG can significantly improve accuracy when the clients can establish a stable collaboration through mutual-correlated data distributions. Take MNIST as an example, we see that clients 0–499 have higher accuracy than all the other clients because their data are beneficial to each other.

Furthermore, if we compare the accuracy of the first 500 clients, we can see that C-DDPG has higher accuracy than FedAvg. The reason is that the RL algorithm in our proposed C-DDPG adaptively increases the weights of the first 500 clients which in a stable coalition. Clients 500–949 have lower accuracy because after an everlasting attempt by our self-organized process, they still cannot collaborate well with others. To protect the global model's performance gains, their weight will be reduced plan they are not in the final stable coalition. It also indicates the reason why the adversary (user ID: 950–999) have reduced accuracy due to its limited mutual beneficial relationship with other clients.

The results derived from the other two data sets have the same observation (Fig. 4). By using C-DDPG, we observe a 3%–12% difference of performance between clients within and outside of the final stable collaboration. Compared with FegAvg, the clients inside the coalition can have better performance gains. For clarity, we list the average accuracy of clients within and outside of this collaboration in Table I.

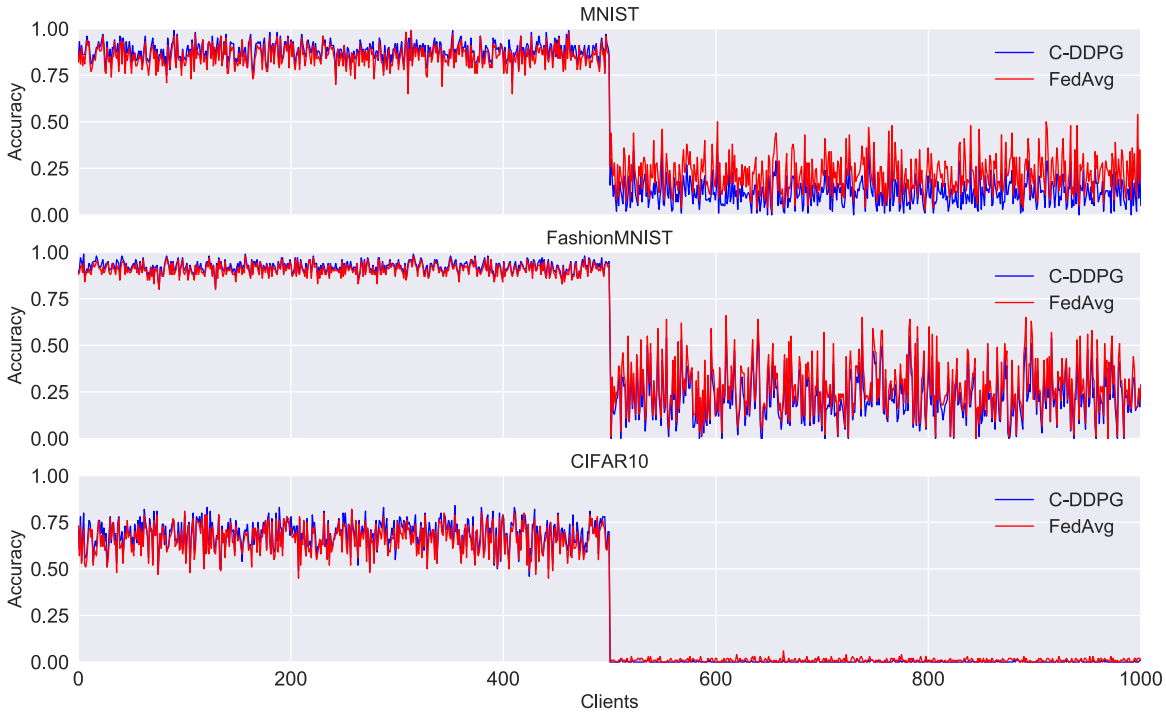


Fig. 4. Accuracy comparison between proposed method and FedAvg by using different data sets.

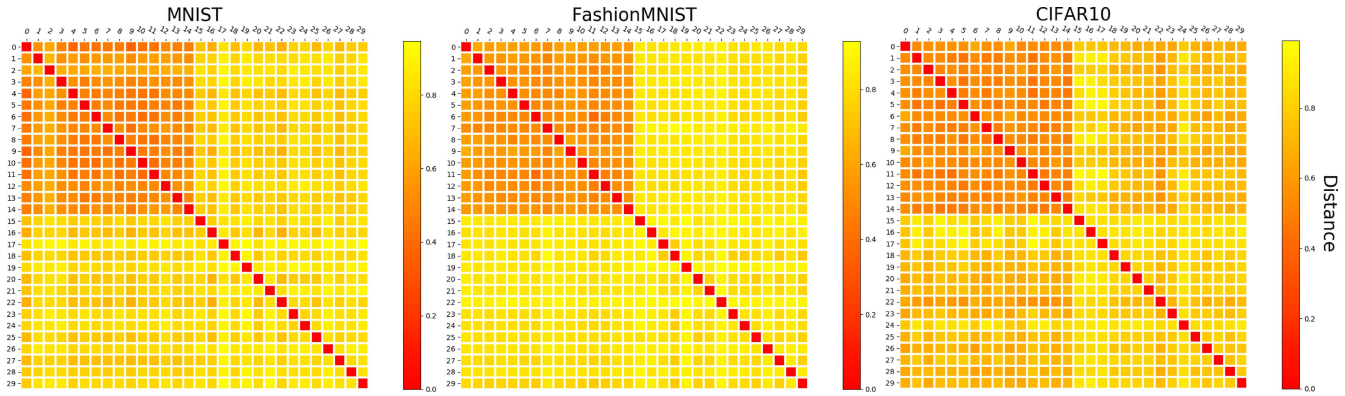


Fig. 5. Correlation comparison between clients of the coalition and clients out of the coalition.

C. Effectiveness on Capturing the Correlation Among Clients

To verify that C-DDPG can effectively recognize the clients has closer correlations and enhance their relationship, we randomly sample 30 uses from three data sets and measure their correlations by two common distance measurement method. Fifteen users are picked from the generated optimal coalition, and the other 15 users are picked from the rest of them. Euclidean distance and Hamming distance are adopted to derive a distance matrix (Fig. 5). The deeper color indicates a small averaged distance or a higher similarity between users. In Fig. 5, we can observe that the top-left area (first 15 users) share the significant deeper shades in all the three data sets while other ones have relatively lighter shades. It validates that our method can ensure the high-related clients are in the stable collaboration plan for all three data sets. Users 25–29 form an area with deep color and failed to join the stable collaboration because of the low similarity with the users in the coalition. Such results indicate the results of our method are quite consistent with the optimal coalition.

D. Accuracy Analysis With Different Heterogeneity Level

In this part, we compare our proposed aggregation process with FedAvg in terms of average accuracy through different heterogeneity level to validate a stable performance in different heterogeneity level settings. To obtain various heterogeneity levels, the ratio of clients that has closer correlations is changed from 100% to 0%.

Fig. 6 represents the trend of averaged accuracy for all situations. We observe that FedAvg and our method have very similar performance when the ratio changes from 0% to 40%. As the heterogeneity level continues to increases (from 40% to 60%), the performance of FedAvg drops while ours still stays high. When the heterogeneity level increases to 40%, the performance difference between FedAvg and C-DDPG starts

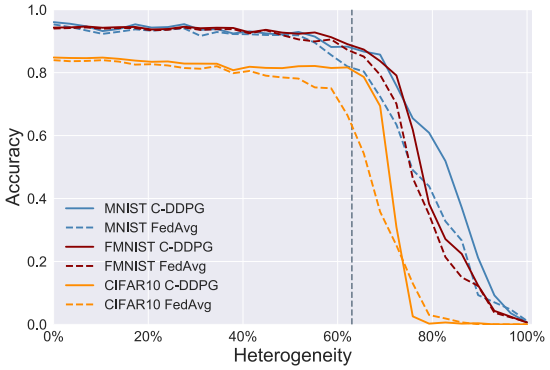


Fig. 6. Heterogeneity versus accuracy.

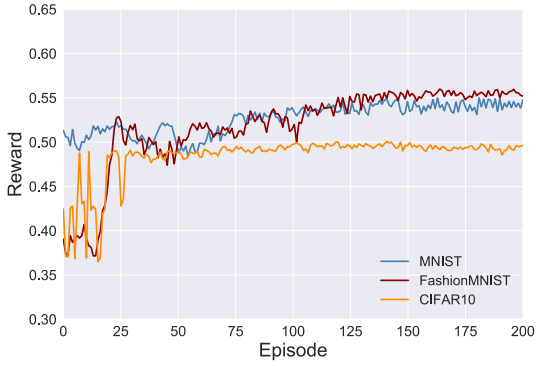


Fig. 7. Cumulative reward versus the number of episodes.

TABLE II
ACCURACY FOR A GIVEN HETEROGENEITY

Dataset	**Objective**	FedAvg (Average)	C-DDPG (Average)
MNIST	inner-coalition	81.01%	**87.16%**
	others	31.89%	**13.83%**
Fashion_MNIST	inner-coalition	86.31%	**88.22%**
	others	35.61%	**8.72%**
CIFAR10	inner-coalition	62.75%	**80.15%**
	others	5.94%	**0.27%**

to increase, and the performance gap becomes more apparent until the heterogeneity reaches around 63%. Table II shows the average accuracy difference with a heterogeneity level as 63%, refer to the dashed line in Fig. 6. For CIFAR10, the average accuracy of clients within a coalition by C-DDPG is 17.4% higher than that by FedAvg. Moreover, the average accuracy of clients outside of a stable coalition by C-DDPG is 5.67% lower than that by FedAvg. Considering that the clients within a stable coalition are the majority of clients, our model improves the accuracy for most clients significantly. When the heterogeneity continuously increases to more than 70%, the performance difference between FedAvg and C-DDPG decreases. One possible reason is that there are too many diverse coalitions generated, which makes the optimization goal blurred. We leave the solution to this "multiple coalitions" phenomenon as our future work.

E. Converge Analysis

Fig. 7 plots the cumulative reward of the proposed algorithm that converged under existing discount factor $\gamma = 0.99$. We only show 200 episodes in the figure since the learned model of all the three data sets becomes stable after that. The fluctuations in the reward curve are because of the noise settings in C-DDPG, which is a method to guarantee the exploration capability during the training process. This result clearly states that the proposed algorithm can converge in a short time.

VI. CONCLUSION

We proposed a novel FL aggregation method. The work first adopts the concept of the self-organizing process and implements such an idea by RL. Our solution has several advantages. First, it can identify the heterogeneity level automatically by testing different collaboration plans during aggregation, and it can also find the consensus clients to form a coalition. Second, an RL-based solution is proposed for weight allocation considering the coalition. Extensive simulations with real data sets reveal that the proposed framework significantly improves the performance of the majority of clients. Despite the initial progress on FL heterogeneity made by this framework, many challenges remain open. In the future, we plan to extend our method for the multicoalition application scenarios.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[2] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6492–6499, Dec. 2019.

[3] S. Kumar and M. Singh, "Big data analytics for healthcare industry: Impact, applications, and tools," *Big Data Min. Anal.*, vol. 2, no. 1, pp. 48–57, 2019.

[4] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 94–101, Feb. 2018.

[5] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.

[6] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[7] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Netw.*, vol. 32, no. 4, pp. 8–14, Jul./Aug. 2018.

[8] L. Liu, X. Chen, Z. Lu, L. Wang, and X. Wen, "Mobile-edge computing framework with data compression for wireless network in energy Internet," *Tsinghua Sci. Technol.*, vol. 24, no. 3, pp. 271–280, 2019.

[9] F. Alam, R. Mehmood, I. Katib, N. N. Albogami, and A. Albeshri, "Data fusion and IoT for smart ubiquitous environments: A survey," *IEEE Access*, vol. 5, pp. 9533–9554, 2017.

[10] L. Deng, M. Yang, Z. Liang, Y. He, and C. Wang, "Fusing geometrical and visual information via superpoints for the semantic segmentation of 3d road scenes," *Tsinghua Sci. Technol.*, vol. 25, no. 4, pp. 498–507, 2020.

[11] J. Liu *et al.*, "Lattice-based double-authentication-preventing ring signature for security and privacy in vehicular ad-hoc networks," *Tsinghua Sci. Technol.*, vol. 24, no. 5, pp. 575–584, 2019.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2017, pp. 1273–1282.

[13] Z. Chai *et al.*, "Towards taming the resource and data heterogeneity in federated learning," in *Proc. USENIX Conf. Oper. Mach. Learn. (OpML)*, 2019, pp. 19–21.

[14] Q. Wu, K. He, and X. Chen. (2020). *Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework*. [Online]. Available: https://arxiv.org/abs/2002.10671

[15] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.

[16] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[17] M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 117–1180.

[18] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 4615–4625.

[19] T. P. Lillicrap *et al.* (2015). *Continuous Control With Deep Reinforcement Learning*. [Online]. Available: https://arxiv.org/abs/1509.02971

[20] C. Wang, C. Wang, Z. Wang, X. Ye, J. X. Yu, and B. Wang, "DeepDirect: Learning directions of social ties with edge-based network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2277–2291, Dec. 2019.

[21] S. Srivastava, G. Ben-Yosef, and X. Boix, "Minimal images in deep neural networks: Fragile object recognition in natural images," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–18.

[22] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI) 30th Innov. Appl. Artif. Intell. (IAAI) 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, 2018, pp. 2836–2843.

[23] J. Son, D. Kim, M. Z. A. Bhuiyan, R. Tashakkori, J. Seo, and D. H. Lee, "Privacy enhanced location sharing for mobile online social networks," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 2, pp. 279–290, Apr.–Jun. 2020.

[24] D. Peteiro-Barral and B. Guijarro-Berdiñas, "A survey of methods for distributed machine learning," *Progr. AI*, vol. 2, no. 1, pp. 1–11, 2013.

[25] Y. Chang, X. Huang, Z. Shao, and Y. Yang, "An efficient distributed deep learning framework for fog-based IoT systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[26] T. Panayiotou, G. Savva, I. Tomkos, and G. Ellinas, "Centralized and distributed machine learning-based QoT estimation for sliceable optical networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–7.

[27] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Min. Anal.*, vol. 3, no. 2, pp. 85–101, 2020.

[28] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, "Clinical big data and deep learning: Applications, challenges, and future outlooks," *Big Data Min. Anal.*, vol. 2, no. 4, pp. 288–305, 2019.

[29] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *Proc. 39th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 144–153.

[30] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 665–673, Jan. 2018.

[31] A. Cotter *et al.*, "Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals," *J. Mach. Learn. Res.*, vol. 20, pp. 1–59, Nov. 2019.

[32] B. Radunovic and J. Le Boudec, "A unified framework for max–min and min–max fairness with applications," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1073–1083, Oct. 2007.

[33] A. Cotter *et al.* (2018). *Optimization With Non-Differentiable Constraints With Applications to Fairness, Recall, Churn, and Other Goals*. [Online]. Available: https://arxiv.org/abs/1809.04198

[34] H. Yu *et al.*, "A fairness-aware incentive scheme for federated learning," in *Proc. AAAI/ACM Conf. AI Ethics Soc.*, 2020, pp. 393–399.

[35] L. U. Khan *et al.* (2019). *Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism*. [Online]. Available: https://arxiv.org/abs/1911.05642

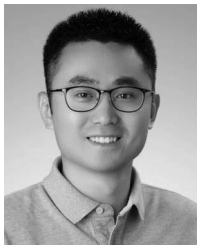
[36] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim. (2019). *Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach*. [Online]. Available: https://arxiv.org/abs/1905.07479

[37] F. Heylighen and C. Gershenson, "The meaning of self-organization in computing," *IEEE Intell. Syst.*, vol. 18, no. 4, pp. 1–6, May/Jun. 2003.

Junjie Pang received the M.S. and Ph.D. degrees from the Department of Computer Science, Jilin University, Changchun, China, in 2013 and 2017, respectively.

He currently holds a Postdoctoral position with Qingdao University, Qingdao, China. His research interests include federated learning, reinforcement learning, and next-generation networking.

Dr. Pang is a recipient of the Initiative Postdocs Program of Shandong Province.



Yan Huang received the Ph.D. degree from the Department of Computing Science, Georgia State University, Atlanta, GA, USA, in 2019.

He is currently an Assistant Professor with the Department of Software Engineering and Game Development, Kennesaw State University, Atlanta. He is broadly interested in privacy and security, with particular emphasis on deep learning aided privacy protection solutions and cybersecurity challenges in the IoT environment. His current research agenda focuses on improving the FL's performance and efficiency, and alleviating the contradictions between multiple training tasks.

Zhenzhen Xie received the M.S. degree in computer science from Jilin University, Changchun, China, in 2014, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Technology.

Her research areas are reinforcement learning, Internet of Things, and representation learning.

Qilong Han received the Ph.D. degree in computer science from Harbin Institute University, Harbin, China, in 2006.

He is a Professor and the Deputy Dean with the College of Computer Science and Technology, Harbin Engineering University, Harbin. He has more than 60 publications as edited books and proceedings, invited book chapters, and technical papers in refereed journals and conferences. His research interests include data security and privacy, mobile computing, distributed, and networked systems.

Prof. Han has served as a Programme Committee Members and the Co-Chairs of a number of international conferences/workshops for areas, including *Web Intelligence*, e-Commerce, Data Mining, and the *Journal of Intelligent Systems*. He is a Senior Member of CCF and the Chair of CCF YOCSEF Harbin.

Zhipeng Cai (Senior Member, IEEE) received the B.S. degree from Beijing Institute of Technology, Beijing, China, and the M.S. and Ph.D. degrees with the Department of Computing Science, University of Alberta, Edmonton, AB, Canada, in 2008.

He is currently an Assistant Professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. He has published more than 50 journals papers, including more than 20 IEEE/ACM Transactions papers, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE/ACM TRANSACTIONS ON NETWORKING, and the IEEE TRANSACTIONS ON MOBILE COMPUTING. His research areas focus on networking, privacy, and big data.

Dr. Cai is a recipient of NSF CAREER Award. He is an Editor/Guest Editor for *Algorithmica*, *Theoretical Computer Science*, the *Journal of Combinatorial Optimization*, and the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS.