Reputation-Aware Hedonic Coalition Formation for Efficient Serverless Hierarchical Federated Learning

Jer Shyuan Ng, Wei Yang Bryan Lim, Zehui Xiong, *Member, IEEE*, Xianbin Cao, *Senior Member, IEEE*, Jiangming Jin, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, Cyril Leung, and Chunyan Miao

Abstract—Amid growing concerns on data privacy, Federated Learning (FL) has emerged as a promising privacy preserving distributed machine learning paradigm. Given that the FL network is expected to be implemented at scale, several studies have proposed system architectures towards improving the network scalability and efficiency. Specifically, the Hierarchical FL (HFL) network utilizes cluster heads, e.g., base stations, for the intermediate aggregation and relay of model parameters. Serverless FL is also proposed recently, in which the data owners, i.e., workers, exchange the local model parameters among a neighborhood of workers. This decentralized approach reduces the risk of a single point of failure but inevitably incurs significant communication overheads. To achieve the best of both worlds, we propose the Serverless Hierarchical Federated Learning (SHFL) framework in this article. The SHFL framework adopts a two-layer system architecture. In the lower layer, the FL workers are grouped into clusters under cluster heads. In the upper layer, the cluster heads exchange the intermediate parameters with their one-hop neighbors without the aid of a central server. To improve the sustainable efficiency of the FL system while taking into account the incentive design for workers' marginal contributions in the system, we propose the reputation-aware hedonic coalition formation game in this article. Specifically, the workers are rewarded for their marginal contribution to the cluster, whereas the reputation opinions of each cluster head is updated in a decentralized manner, thereby deterring malicious behaviors by the cluster head. This improves the performance of the network since cluster heads with higher reputation scores are more reliable in relaying the intermediate model parameters. The simulation results show that our proposed hedonic coalition formation algorithm converges to a Nash-stable partition and improves the network efficiency.

Index Terms—Federated learning, serverless federated learning, decentralized edge intelligence, hedonic coalition formation

- *Jer Shyuan Ng and Wei Yang Bryan Lim are with Alibaba Group and Alibaba-NTU Joint Research Institute (JRI), Nanyang Technological University (NTU), Singapore 639798. E-mail: {s190068, limw0201}@e.ntu.edu.sg.*
- *Zehui Xiong is with the Singapore University of Technology and Design (SUTD), Information Systems Technology and Design (ISTD) Pillar, Singapore 639798. E-mail: zehui_xiong@sutd.edu.sg.*
- *Xianbin Cao is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China. E-mail: xbcao@buaa.edu.cn.*
- *Jiangming Jin is with TuSimple, Beijing 100016, China. E-mail: jiangming.jin@outlook.com.*
- *Dusit Niyato is with the School of Computer Science and Engineering (SCSE), NTU, Singapore 639798. E-mail: dniyato@ntu.edu.sg.*
- *Cyril Leung is with The University of British Columbia, Vancouver, BC V6T 1Z4, Canada, and also with Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Singapore 639798. E-mail: cleung@ece.ubc.ca.*
- *Chunyan Miao is with SCSE, NTU, Singapore, Alibaba-NTU JRI, LILY, Singapore 639798. E-mail: ascymiao@ntu.edu.sg.*

Manuscript received 27 Aug. 2021; revised 21 Dec. 2021; accepted 23 Dec. 2021. Date of publication 29 Dec. 2021; date of current version 23 May 2022.
This work was supported in part by the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, in part by the Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), in part by the National Research Foundation, Singapore under the AI Singapore Programme under Grant AISG2-RP-2020-019, in part by the WASP/NTU under Grant M4082187 (4080), in part by the Singapore Ministry of Education (MOE) Tier 1 under Grant RG16/20, in part by the SUTD under Grant SRG-ISTD-2021-165, in part by the SUTD-ZJU IDEA under Grant SUTD-ZJU (VP) 202102, in part by the SUTD-ZJU IDEA Seed under Grant SUTD-ZJU SD 202101, and in part by the NSFC under Grants 61827901 and 62071343.
Corresponding author: Zehui Xiong.
Recommended for acceptance by A. J. Peña, M. Si, and J. Zhai.
Digital Object Identifier no. 10.1109/TPDS.2021.3139039

1 Introduction

With the advent of 5G and the enhanced perception and communication capabilities of the Internet of Things (IoT), it has been envisioned that the IoT will continue to grow in terms of its influence in multiple facets of society, e.g., the Internet of Medical Things (IoMT) has paved the way towards Medicine 4.0 [1], similar to the role of Industrial IoT in Industry 4.0 [2].

Coupled with the rise of Artificial Intelligence (AI), the large volumes of data captured by sensors deployed in the IoT can contribute to the development of effective machine learning models. Traditionally, the training data across different data owners are pooled in a central cloud server for model training. However, this approach has been restricted by increasingly stringent policies, e.g., the Health Insurance Portability and Accountability Act (HIPAA) [3] and the General Data Protection Regulation (GDPR) [4]. With the growing restrictions on sharing training data with external parties, the development and deployment of data-driven AI solutions have been severely impeded.

In face of these challenges, Federated Learning (FL) has been proposed in [5] to enable privacy preserving collaborative machine learning across multiple data owners, i.e., workers. Conventionally, in each iteration of FL (VFL), a model owner transmits a set of model parameters to each worker. Then, the model parameters are updated with the worker's locally stored training data. Only the updated model parameters, rather than the raw data, need to be transmitted back to the model owner for global aggregation. FL has found recent successes in applications such as next word prediction model for the Google Keyboard [6], in healthcare [7], [8], and environment monitoring [9].

The FL network is envisioned to involve thousands of heterogeneous distributed devices [4]. Given that synchronous cloud-aggregation remains the predominant approach for global model aggregation in FL, node failures, communication bottlenecks, and the single point of failure due to the computation and communication inefficiencies have proven to be key bottlenecks that impede the effective and scalable implementation of FL [10].

As a solution, edge computing and cooperative communications inspired approaches have recently been proposed. The Hierarchical FL (HFL) framework proposed in [11] adopts a partially decentralized architecture in which workers do not communicate directly with a centralized model owner, e.g., the macro cell base station of a model owner. Instead, the local model parameters are first uploaded to edge servers, e.g., micro base stations, for intermediate parallel edge aggregation. At prespecified intervals, the edge servers relay the edge-aggregated intermediate parameters to the model owner for global aggregation. This approach reduces the instances of costly cloud aggregation and mitigates disruptive device dropouts that can happen due to end devices that suffer from poor network conditions. However, the HFL approach still relies on a central server and therefore has the drawback of a single point of failure and performance bottleneck. On the other end of the spectrum, the Serverless FL (SFL) framework proposed in [12] adopts a fully decentralized approach in which the workers form neighborhoods to exchange their model parameters, thereby relying on device-to-device (D2D) communication and peer-to-peer networking to eliminate the need for intermediate or centralized aggregation. However, the communication overheads increase quickly with the number of workers, therefore limiting the scalability of the SFL network.

To capitalize benefits of HFL and SFL, we propose the Serverless Hierarchical Federated Learning (SHFL) framework in this paper. The SHFL framework adopts a two-layer system architecture to reduce communication overheads while alleviating the reliance on a central server for model aggregation. Systems with many layers cause more communications and aggregation overhead. Two-layer architecture is commonly adopted in systems to strike a balance between performance and overheads [13]. In the lower layer, the FL workers are grouped into clusters under cluster heads, e.g., edge servers, similar to that of HFL. The cluster heads serve to perform intermediate aggregation of the local model parameter updates. In the upper layer, the cluster heads exchange the aggregated intermediate parameters with their one-hop neighbors without the aid of a central server, similar to that of SFL.

The decentralized aggregation eliminates the single point of failure and performance bottleneck but renders the SHFL system to be vulnerable to more attack surfaces or free riding attacks. For example, a malicious cluster head may implement model poisoning attacks [14] that deteriorate the performance of the global model. A cluster head may also free ride on the efforts of other cluster heads, e.g., by delaying the intermediate model aggregation and transmission to save on resource cost. To improve the sustainable efficiency of the SHFL network, we propose a reputation system that takes into account the positive and negative interactions among the cluster heads. The reputation scores of the cluster heads are known to other cluster heads and all FL workers. The FL workers utilize this knowledge of the reputation scores of the cluster heads to decide on which cluster head to join. The reputation scores are updated in a decentralized manner and penalize negative interactions heavily, thereby deterring erroneous behaviors by the cluster heads.

In addition, the issue of incentivizing the non-cooperative workers to participate in the model training is not well-addressed. In contrast to vanilla FL systems, the workers may choose to join different cluster heads for the FL training, rather than a single model owner. To motivate the FL workers' participation in model training, the FL workers receive a reward from the cluster head that it joins for its marginal contribution to the cluster. Besides, as the reward pool is shared by the FL workers in the same coalition, the share of reward pool received by each FL worker is reduced by the addition of a new member in the coalition. As a result, some FL workers may leave their current coalition and join another coalition to increase their utility. Given that the objective of the non-cooperative FL workers is to maximize their own utilities, regardless of the effect of their decisions on the utilities of other FL workers, we propose the reputation-aware hedonic coalition formation game in this paper.

In summary, our key contributions are as follows:

1) We propose a Serverless Hierarchical Federated Learning (SHFL) framework that alleviate the need of a central server while reducing the communication costs of the FL training.
2) We highlight the use of the reputation score of the cluster heads to signal the reliability of the cluster heads in performing the relay of the intermediate model parameters where cluster heads with higher reputation scores tend to attract more FL workers to join their clusters.
3) We formulate the cluster formation problem as a hedonic coalition formation game to model the incentivization of the workers and the cluster formation in the SHFL network. The hedonic coalition formation game captures the selfish behaviour of the FL workers in maximizing their own utilities without consideration of the membership of the cluster. We also show that the hedonic coalition formation algorithm converges to a Nash-stable partition.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 discusses the system model. Sections 4 and 5 present the hedonic coalition formation among the FL workers and simulation results respectively. Section 6 concludes the paper.

2 Related Work

2.1 Federated Learning

FL is a privacy-preserving machine learning paradigm first proposed in [5]. In distributed learning schemes such as FL, the communication cost often dominates the computation cost. Several works have proposed a variety of solutions, e.g., model compression techniques such as quantization [15] and subsampling [16], client selection protocols to reduce the occurrences of stragglers [17], robust design to reduce the effect of noise for FL over noisy channels [18] and to improve energy efficiency [19], as well as Broadband Analog Aggregation (BAA) with over-the-air computation to enable the reuse of the whole bandwidth for scalable FL [20].

Recently, edge computing-inspired solutions have been proposed to further enhance the communication efficiency of FL [21]. In [11], the HFL framework proposed involves workers transmitting the updated local model parameters to small-cell base stations for intermediate aggregation. To ensure efficient HFL, a socially-aware cluster head selection algorithm is proposed in [22] in which the cluster head is chosen based on its relationship with the workers. In [19], instead of small-cell base stations, Unmanned Aerial Vehicles are utilized to to provide relay support and intermediate aggregation for efficient FL. The UAVs can be flexibly deployed the cells which are distant from the FL server, thereby reducing communication latency.

In addition to the above measures, various networking topologies have been proposed to fulfill different FL training tasks. In [12], the SFL architecture is proposed. The serverless aspect of FL removes the reliance of the workers on a central server, thereby eliminating the single point of failure. Due to the significant communication overhead, a consensus based federated averaging with gradient exchange strategy is proposed. Specifically, the local gradients and model updates are jointly exchanged among workers with their neighbors in a decentralized manner. Nevertheless, the communication cost is still high and impedes the network scalability.

Motivated by the above studies, we propose the SHFL approach in this paper to achieve the best of both worlds of communication cost reduction without the drawback of a single point of failure.

2.2 Reputation and Coalition Formation in Federated Learning

Despite the advantages of FL in enabling collaborative model training, a major problem is that FL involves heterogeneous workers and cluster heads. For example, the workers and cluster heads differ in terms of computation and communication capabilities [23], as well as willingness to participate in the FL training. In addition, malicious workers or cluster heads may risk the privacy of the workers involved or lower the quality of the global model through data poisoning or model poisoning [24], [25].

In response, mechanism design tools have been utilized to improve the efficiency of an FL network. The study in [26] proposes a contract theoretic and coalition formation game as a mechanism to incentivize FL and mitigate free riding. Amid information asymmetry, the incentive compatibility aspect of contract theory devises rewards such that the workers are rewarded based on their intrinsic costs of FL participation. Moreover, the coalition formation game accords rewards to each participant in the FL training its marginal contribution to the coalition, thereby mitigating free riding.

In some cases, FL is carried out over a prolonged period where data is continuously collected and the model is improved [2]. As such, the study of [27] proposes a reputation blockchain to store the reputation scores of the FL workers based on their contribution to the model training process each time period. An FL worker with malicious behavior is penalized through reputation score deduction. This ensures that adversarial workers (with lower reputation) are omitted from the FL training. The study of [28] explores the domain reputation and content characteristics of the fake and real news in order to detect the fake news on social media. In [29], the hierarchical game framework of evolutionary game and auction is proposed for HFL network optimization. The evolutionary game models the self-organizing aspects of cluster formation in HFL, whereas the deep learning based auction guarantees the desirable properties of profit-maximization of the auctioneer and incentive-compatibility of the seller.

3 System Model

We consider an FL network with M edge servers, e.g., base stations, serving as cluster heads, and J IoT devices, i.e., FL workers, which are represented by the sets $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$ and $\mathcal{J} = \{1, \ldots, j, \ldots, J\}$ respectively. The cluster heads and workers are located across the network at the coordinates (x_m, y_m) and (x_j, y_j) respectively.

Each worker has d_j data points represented by $\left(\mathbf{X}^j, \mathbf{Y}^j\right)$ where

$$\mathbf{X}^j = \begin{pmatrix} \mathbf{x}_1^j \\ \vdots \\ \mathbf{x}_{d_j}^j \end{pmatrix}, \quad \mathbf{Y}^j = \begin{pmatrix} y_1^j \\ \vdots \\ y_{d_j}^j \end{pmatrix},$$

and $\mathbf{x}_z^j$ is a vector of features associated with the scalar label y_z^j for $z = \left\{1, \ldots, d_j\right\}$. The workers aim to build a global model collaboratively without the exposure of their private data. The collaborative model training can be implemented over K iterations represented by $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ through the following frameworks:

- *Vanilla FL:* In iteration k of Vanilla FL, each FL worker receives a set of global model parameters denoted $\boldsymbol{w}^{(k)}$ from a *central* model owner. The FL worker j derives the updated local model parameters $\boldsymbol{w}_j^{(k+1)}$ through model training over its local data to minimize the training loss $L_j(\boldsymbol{w}_j)$. Thereafter, the local parameters $\boldsymbol{w}_j^{(k+1)}, \forall j \in \mathcal{J}$, are transmitted to the model owner for global aggregation to derive $\boldsymbol{w}^{(k+1)}$, e.g., through the *FedAvg* algorithm [5].
- *Hierarchical FL:* The HFL framework involves cluster heads as an addition. The cluster heads serve to perform intermediate aggregation and the relay of model parameters to the model owner. This reduces communication latency and instances of device dropouts [11]. The FL workers form M clusters, each of which is in turn associated with a cluster head, e.g., through socially-aware clustering approaches [22] or evolutionary schemes [29]. In iteration k, each cluster head m receives a set of global parameters denoted $\boldsymbol{w}^{(k)}$ from the model owner. The FL worker j derives the

updated local model parameter $w_j^{(k+1)}$ through model training. Thereafter, the local parameters $w_j^{(k+1)}$ are transmitted to the workers' respective cluster heads for *intermediate* aggregation to derive $w_m^{(k+1)}$. The intermediate parameters are transmitted to the workers in the cluster for more iterations of updating. At predefined intervals κ, the intermediate parameters $\forall m \in \mathcal{M}$ are transmitted to the model owner for global aggregation to derive $w^{(k+\kappa)}$, which will be used for the subsequent transmission to the workers in the clusters.

- *Serverless FL:* The SFL framework is fully decentralized, and is conducted through D2D communications without the cluster head or central model owner [12]. The FL workers are segmented into M clusters or neighborhoods. In iteration k, the FL worker j in cluster m receives the model parameters from all its neighbors in $m\backslash\{j\}$. The decentralized aggregation is conducted on each worker to derive the global model parameter $w_m^{(k)}$, which is subsequently used for local model training by the worker.

To achieve the best of HFL and VFL, we propose the Serverless Hierarchical FL (SHFL) (Fig. 1) in this paper. The SHFL framework adopts a two-layer system to *both* reduce communication latency and alleviate the reliance on a central server for model aggregation. In SHFL, each k^{th} iteration consists of four steps:

1) *Local Computation:* The FL workers receive the intermediate FL model from the cluster heads, i.e., $w_m^{(k)}$, for model training on their datasets locally.
2) *Transmission of Local Model Parameters:* The FL workers transmit the updated model parameters $w_j^{(k+1)}$ to the cluster heads.
3) *Intermediate Aggregation:* The cluster heads aggregate the local model parameters to derive the intermediate model $w_m^{(k+1)}$, which is transmitted back to the FL workers for the $(k+1)^{\text{th}}$ iteration.
4) *Global Aggregation:* At the end of the prespecified interval κ, each cluster head transmits the intermediate model parameters $w_m^{(k+\kappa)}$ to their one-hop neighbours.[1] At the same time, they receive the intermediate model parameters of their neighbouring cluster heads $w_{m'}^{(k+\kappa)}$. The received intermediate model parameters are aggregated to update the global FL model to derive $w^{(k+\kappa)}$. Given the updated global FL model, the cluster heads conducts the next interval of κ iterations of FL training with the FL workers in the clusters.

In terms of communication cost, the communication cost in SHFL is larger than that in HFL, but smaller than that in SFL. Due to the exchange of aggregated intermediate parameters between the cluster heads in the upper layer of the SHFL framework, the communication cost in SHFL is larger than that in HFL. Besides, in SHFL, the exchange of model parameters is performed between the cluster heads whereas in SFL, the exchange of model parameters is performed between the FL workers. Typically, FL workers are in much larger number than the cluster heads. Hence, the number of model parameter exchanges and thus the communication cost are smaller in SHFL than that in SFL.

1. In the experimental results of Reference [12], the authors have shown that the average execution time for model convergence is greatly reduced, even when only one-hop neighborhood is used. The increase in the number of hops for the exchange of the aggregated intermediate parameters may increase the communication cost greatly, which is not feasible for a large FL network with hundreds or thousands of devices.

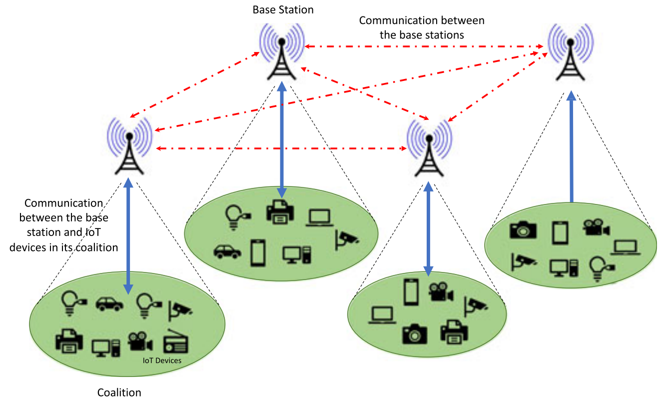


Fig. 1. Serverless Hierarchical Federated Learning (SHFL) framework.

In terms of handling the issue of workers failure due to poor or unstable Internet connection, the performance of SHFL and HFL frameworks are similar, but they are both better than the SFL framework. The lower layer of the SHFL framework is similar to that of the Hierarchical Federated Learning (HFL) where the local model parameters are first uploaded to the edge servers for intermediate parallel edge aggregation. This approach does not only reduce the instances of costly cloud aggregation, but also mitigate disruptive device dropouts that can happen, e.g., due to end devices that suffer from poor network conditions. Since the lower layer of the SHFL framework is similar to that of the FL framework, the performances of SHFL and HFL frameworks in handling large number of workers, devices dropout and failures are the same. The aim of the SFL framework is to avoid a single point of failure by alleviating the need for a central server. The SFL framework does not address the issue of devices dropout and failures. In particular, the larger the number of devices that fail to transmit their local model parameters, the longer the time required for convergence. As such, due to the intermediate parallel edge aggregation in the lower layer of SHFL, the SHFL framework is better in handling the issue of devices dropout and failures than the SFL framework.

To improve the performance of its intermediate model, each of the cluster heads has the objective of attracting more FL workers which increases the quantity and also quality in terms of diversity of data used in the FL training. To encourage the participation of the FL workers, each of the cluster heads offers a reward pool to be shared among the FL workers in its cluster. Since the FL workers are only able to facilitate the FL training of a single cluster head, the FL workers choose to join the clusters that maximizes their utilities. On one hand, the FL workers are attracted to join the cluster head with a high reward pool. On the other hand, when there are more FL workers join that cluster, the proportion of reward that each FL worker receives is smaller. To derive the cluster formation among the FL workers, we adopt the hedonic coalition formation game to model the utility-maximizing behaviour of the FL workers.

TABLE 1
System Model Parameters

Parameter	Description
M	Number of cluster heads
J	Number of FL workers
d_j	Number of data points
K	Number of iterations
$v(S_m)$	Value of coalition S_m
γ_m	Reputation score
$b^t_{\tilde{\mathcal{M}}\to m}$	Belief scores
$d^t_{\tilde{\mathcal{M}}\to m}$	Disbelief scores
$u^t_{i,m}$	Uncertainty scores
$\alpha^t_{i,m}$	Positive interactions between cluster heads
$\beta^t_{i,m}$	Negative interactions between cluster heads
ζ	Weight of positive interactions
η	Weight of negative interactions
$q^t_{i,m}$	Packet error rate
ϑ_t	Freshness fading function
z	Fade parameter
ω	Influence of uncertainty on reputation score
$\rho_j^{S_m}$	Payoff of FL worker j from joining coalition S_m
c_j^{cp}	Computation cost of FL worker j
θ_1	Unit computation cost
μ	Computational coefficient
a_j	Total number of CPU cycles
f_j	Computational capability of FL worker j
$r_{j,m}$	Transmission rate
P_j	Transmission power
B_m	Resource allocated to FL worker by cluster head m
N_0	Noise power spectral density
$h_{j,m}$	Channel gain
$c^{cm}_{j,m}$	Communication cost
θ_2	Unit communication cost
W	Size of local model parameters
β_m	Congestion coefficient
$z_j^{S_m}$	Disutility of FL worker j from joining coalition S_m
$x_j^{S_m}$	Utility fo FL worker j from joining coalition S_m
Pi	Coalitional structure
$u_j^{S_m}$	Preference function of FL worker j
$h(j)$	History set of FL worker j

4 Hedonic Coalition Formation

In this section, we first introduce the reputation, reward, and cost modeling to formulate the worker utilities. Then, the hedonic coalition game is introduced together with the hedonic coalition formation algorithm to derive the worker coalitions. In hedonic games, the FL workers, which are self-interested and aim to maximize their own utilities, have preference over which cluster they want to join. As such, their utilities are only affected by the members in the coalitions to which they belong. This is different from the generic coalitional games in [30] and [31] that aim to maximize the value of the coalitions. A list of the system model parameters used is given in Table 1.

4.1 Worker Reward, Reputation, and Cost Model

To formulate the worker utilities, we consider that the payoff received by the FL worker j from joining coalition S_m consists of two components: i) the marginal reward for local model contribution, and ii) the utility derived from the reputation score of coalition S_m.

4.1.1 Marginal Reward

The cluster heads aim to form larger coalitions with more FL workers so that they have more training data, which helps to improve the model performance. The value of the coalition S_m, which is associated with cluster head m, is expressed as follows

$$v(S_m) = \sum_{j\in S_m} d_j, \tag{1}$$

where d_j is the data quantity of FL worker j.

To attract the FL workers to facilitate its FL training process, cluster head m offers a reward pool α_m which is to be shared among the FL workers in its cluster. The share of reward pool gained by the FL workers depends on their marginal data contribution $\frac{d_j}{\sum_{j\in S_m} d_j}$. Specifically, the larger the quantity of data contributed during local model training, the larger the share of reward pool that the FL worker gains.

4.1.2 Reputation Model

The utility of the worker is also affected by the reputation γ_m of the cluster head it joins. In general, a cluster head with higher reputation score provides the positive signaling effect [32] that the cluster head is non-malicious and reliable in performing the intermediate model aggregation and relay.

The reputation score of each cluster head can be calculated using the multiweight subjective logic scheme [27]. The reputation score of cluster head m is maintained by the set $\tilde{\mathcal{M}} = \mathcal{M}\backslash\{m\}$ of $M-1$ cluster heads with which it has interacted. The reputation score is derived using the reputation opinions $\{b^t_{\tilde{\mathcal{M}}\to m}, d^t_{\tilde{\mathcal{M}}\to m}, u^t_{\tilde{\mathcal{M}}\to m}\}$ of $\tilde{\mathcal{M}}$ for cluster head m. The reputation opinions refer to the belief, disbelief, and uncertainty scores respectively for period t and can be derived [33], [34] as follows

$$\begin{cases} b^t_{\tilde{\mathcal{M}}\to m} = \sum_{i\in\tilde{\mathcal{M}}} u^t_{i,m} \dfrac{\zeta\alpha^t_{i,m}}{\zeta\alpha^t_{i,m} + \eta\beta^t_{i,m}}, \\ d^t_{\tilde{\mathcal{M}}\to m} = \sum_{i\in\tilde{\mathcal{M}}} u^t_{i,m} \dfrac{\eta\beta^t_{i,m}}{\zeta\alpha^t_{i,m} + \eta\beta^t_{i,m}}, \\ u^t_{i,m} = 1 - q^t_{i,m}, \end{cases} \tag{2}$$

where $\alpha^t_{i,m}$ and $\beta^t_{i,m}$ refer to the instances of positive and negative interactions between cluster head m and cluster head i for $i \in \tilde{\mathcal{M}}$ respectively. The positive and negative interactions can be determined through identifying adversarial updates, e.g., using the *FoolsGold* approach [35], or identifying straggling cluster heads that are slow to transmit the intermediate parameters. ζ and η refer to the weights on reputation calculation for the positive and negative interactions respectively. For example, $\eta > \zeta$ refers to the case in which negative interactions are penalized more heavily. $q^t_{i,m}$ refers to the packet error rate [36], in which a larger bit error rate implies a greater extrinsic uncertainty in computing the reputation opinion.

Each period t denotes $a\kappa$ intervals of local model training where a is a positive integer. To consolidate the reputation opinions across T periods into a single reputation score across all preceding periods, we adopt the freshness fading function

$\vartheta_t = z^{T-t}$ [27], where T refers to the current time period and $z \in (0,1)$ refers to the fade parameter. The freshness fading function ensures that the most recent interactions have larger weights than past events. The consolidated reputation opinions are given as follows

$$\begin{cases} b_{\tilde{\mathcal{M}}\to m} = \dfrac{\sum_{t=1}^{T} \vartheta_t b_{\tilde{\mathcal{M}}\to m}^t}{\sum_{t=1}^{T} \vartheta_t}, \\ d_{\tilde{\mathcal{M}}\to m} = \dfrac{\sum_{t=1}^{T} \vartheta_t d_{\tilde{\mathcal{M}}\to m}^t}{\sum_{t=1}^{T} \vartheta_t}, \\ u_{\tilde{\mathcal{M}}\to m} = \dfrac{\sum_{i\in\tilde{\mathcal{M}}} \sum_{t=1}^{T} \vartheta_t u_{i,m}^t}{\sum_{t=1}^{T} \vartheta_t}. \end{cases} \tag{3}$$

By using the consolidated opinions, the reputation γ_m of cluster head m can be derived by

$$\gamma_m = b_{\tilde{\mathcal{M}}\to m} + \omega u_{\tilde{\mathcal{M}}\to m}, \tag{4}$$

where $\omega \in [0,1]$ refers to the influence of the uncertainty on the reputation score.

Hence, the payoff that is received by FL worker j from joining coalition S_m is given as follows

$$\rho_j^{S_m} = \frac{d_j}{\sum_{j'\in S_m} d_{j'}} \alpha_m + \gamma_m. \tag{5}$$

4.1.3 *Cost Model*

To facilitate the FL training of the cluster heads, the FL workers incurs computation and communication costs. In particular, the computation cost of FL worker j, which is denoted as c_j, is expressed as follows

$$c_j^{cp} = \theta_1 \mu a_j f_j^2, \tag{6}$$

where θ_1 is the unit computation cost, μ is the coefficient of the value that depends on the circuit architecture of the central processing unit (CPU) [37], a_j is the total number of CPU cycles required by FL worker j to perform the FL training, and f_j is the computation capability of FL worker j which is determined by the clock frequency of the CPU.

To facilitate the transmission of local model parameters from the FL workers to the cluster heads, the cluster heads distribute the orthogonal resource blocks to the FL workers in their clusters. The transmission rate $r_{j,m}$ from the FL worker to the cluster head can be represented by

$$r_{j,m} = B_m \log_2 \left(1 + P_j h_{j,m}/N_0\right), \tag{7}$$

where P_j denotes the transmit power of the worker j, B_m denotes the resource allocated to each worker by the cluster head, N_0 denotes the noise power spectral density, and $h_{j,m}$ denotes the channel gain from the worker j to edge server m, which is proportional to the distance between the worker and the edge server. Accordingly, the communication cost incurred by FL worker j to transmit the local model parameters of size W is expressed as follows

$$c_{j,m}^{cm} = \theta_2 \frac{W}{r_{j,m}}, \tag{8}$$

where θ_2 is the unit communication cost.

In addition, the limited wireless capacity in the physical communication effects may lead to the congestion effect, which in turn leads to an increase in FL training latency. Specifically, with more FL workers joining a particular cluster, the congestion effect is greater, thereby resulting in disutility among the FL workers. The disutility of FL worker j from joining coalition S_m that results from the congestion effect [38] is modelled as follows

$$z_j^{S_m} = \beta_m \left(\sum_{j\in S_m} r_{j,m} \right)^2, \tag{9}$$

where $\beta_m > 0$ is the congestion coefficient that is determined by the resource constraint of cluster head m and $\sum_{j\in S_m} r_{j,m}$ is the total usage of communication resources of FL workers in coalition S_m.

As such, the utility of FL worker j in coalition S_m, which is denoted as $x_j^{S_m}$, is defined as follows

$$x_j^{S_m} = \rho_j^{S_m} - c_j^{cp} - c_{j,m}^{cm} - z_j^{S_m}. \tag{10}$$

Note that the FL workers do not communicate with each other since they make decisions based on their own utilities, regardless of the effect of their decisions on the utilities of other FL workers.

4.2 Hedonic Coalition Game Formulation

We present the useful definitions in the hedonic coalition game formulation as follows.

Definition 1. *A coalition of FL workers is denoted as $S_m \subseteq \mathcal{J}$ where m is the index of the cluster head.*

Definition 2. *The set $\Pi = \{S_1, \ldots, S_m, \ldots, S_M\}$ is a partition or coalitional structure of $\mathcal{M}$ that spans all FL workers in $\mathcal{J}$, where $S_m \cap S_{m'} = \emptyset$ for $m \neq m'$, $\bigcup_{m=1}^{M} S_m = \mathcal{J}$, and M is the total number of coalitions in partition Π.*

Since there are M cluster heads, there are M coalitions, where each FL worker can choose to join and facilitate the FL training of one of the M cluster heads. $\mathcal{J}$ denotes the coalition of all FL workers, which is also referred to as the grand coalition. The formation of a grand coalition means that all FL workers facilitate the FL training of a single cluster head. A singleton coalition refers to a coalition that contains a single FL worker that facilitates the FL training of the cluster head. If no FL worker is willing to facilitate the FL training of a cluster head, the coalition associated with that cluster head is represented by an empty set $\emptyset$. Note that an FL worker does not join any of the cluster heads if its utility of joining the cluster heads is negative.

In the hedonic game setting, each FL worker $j \in \mathcal{J}$ needs to build preferences over all possible coalitions that it can join, where each worker j compares the different coalitions and order them based on its preferences. In order to evaluate the preferences of the FL workers over their own sets of possible coalitions, the concept of preference relation is introduced.

Definition 3. *For any FL worker $j \in \mathcal{J}$, a preference relation $\succeq_j$ is defined as a complete, reflexive and transitive binary relation over the set of all coalitions that FL worker j can possibly join.*

In order to compare the preference of FL worker j over different coalitions, the preference relation of FL worker $j \in \mathcal{J}$ is defined as follows

$$S_1 \succeq_j S_2 \Leftrightarrow u_j^{S_1} \geq u_j^{S_2}, \tag{11}$$

where $S_1 \in \mathcal{J}$ and $S_2 \in \mathcal{J}$ are two possible coalitions that FL worker j can join, $u_j^{S_m}$ is the preference function of FL worker $j \in \mathcal{J}$ for coalition S_m, $\forall m \in \mathcal{M}$. In particular, for an FL worker $j \in \mathcal{J}$, given two coalitions $S_1 \in \mathcal{J}$ and $S_2 \in \mathcal{J}$ such that $j \in S_1$ and $j \in S_2$, $S_1 \succeq_j S_2$ indicates that FL worker j prefers to join coalition S_1 over coalition S_2, or at least FL worker j prefers both coalitions equally. Its asymmetric counterpart, which is denotes as $\succ_j$, when used in $S_1 \succ_j S_2$ means that FL worker j strictly prefers to join coalition S_1 over coalition S_2. The preference relation $\succeq_j$ allows the FL workers to quantify their preferences, which can be application-specific. It can be expressed as a function of many parameters, such as the payoffs that the FL workers receive from different coalitions and the proportion of FL workers' contribution in different coalitions.

The preference function of FL worker j in coalition S_m, which is denoted as $u_j^{S_m}$, is defined as follows

$$u_j^{S_m} = \begin{cases} x_j^{S_m}, & \text{if } S_m \notin h(j), \\ -\infty, & \text{otherwise}, \end{cases} \tag{12}$$

where $x_j^{S_m}$ is the utility of FL worker j in coalition S_m defined in (10) and $h(j)$ is the history set of FL worker j. The history set contains the coalitions that FL worker j has joined prior to formation of the current partition Π. The preference function in (12) allows the FL workers to choose a coalition that maximizes their utilities. The FL workers avoid any coalition that it has previously visited. This helps to reduce the complexity of the hedonic coalition formation algorithm since the already-visited coalitions are excluded from the set of choices of the FL workers [39]. Given the preference function of FL worker j, the preference relation can be easily generated by comparing the utility of FL worker j for each pair of coalitions. Note that when the history sets are not used, the hedonic coalition formation algorithm still converges to a stable partition, but at the expense of longer convergence time.

Given the set of FL workers $\mathcal{J}$ and a preference relation $\succeq_j$ for every FL worker $j \in \mathcal{J}$, a hedonic coalition formation game is formally defined as follows:

Definition 4. *A hedonic coalition formation game is a coalitional game that is defined by the pair $(\mathcal{J}, \succ)$, where $\mathcal{J}$ is the set of FL workers and $\succ = \{\succ_1 \cdots \succ_j \cdots \succ_J\}$ is the profile of preferences defined for each FL worker in $\mathcal{J}$. In addition, the hedonic coalitional game satisfies the following two conditions:*

1) *The payoff of any FL worker depends solely on the members of the coalitions to which the FL worker belongs, and*
2) *The coalitions form as a result of the preferences of the FL workers over their sets of possible coalitions.*

Definition 5. *(Switch Rule) Given a partition $\Pi = \{S_1, \ldots, S_m, \ldots, S_M\}$, an FL worker $j \in S_m$ decides to leave it current coalition S_m and join another coalition $S_{m'} \in \Pi$, where $m \neq m'$, if and only if $S_{m'} \bigcup \{j\} \succ_j S_m$. As a result, $\{S_m, S_{m'}\} \rightarrow \{S_m \backslash \{j\}, S_{m'} \bigcup \{j\}\}$.*

Given a partition Π, the switch rule in hedonic coalition formation games provide a mechanism which allows any FL worker j to leave its current coalition S_m and join another coalition $S_{m'} \in \Pi$, where $m \neq m'$, given that the new coalition $S_{m'} \bigcup \{j\}$ is strictly preferred over the current coalition S_m based on the defined preference relation. Therefore, through a single switch rule performed by any FL worker $j \in \mathcal{J}$, the current partition Π of the set of FL workers $\mathcal{J}$ is transformed into a new partition $\Pi' = \Pi \backslash \{S_m, S_{m'}\} \bigcup \{S_m \backslash \{j\}, S_{m'} \bigcup \{j\}\}$. Based on its preference relation, an FL worker leaves its current coalition and joins another coalition, regardless of the effect of its decision on other FL workers. As such, the switch rule in the hedonic coalition formation games reflects the selfish behaviour of the FL workers.

In the hedonic coalition formation games, there exists a stable partition. We present two types of stable partitions as follows [40]:

- *Nash-stable:* A partition $\Pi = \{S_1, \ldots, S_m, \ldots, S_M\}$ is Nash-stable if $S_m \succ_j S_{m'} \bigcup \{j\}$, where $m \neq m'$, $\forall j \in S_m$ and $\forall S_m \in \Pi$. In particular, a partition Π is Nash-stable if there is no FL worker $j \in \mathcal{J}$ has an incentive to leave its current current coalition S_m and join another coalition $S_{m'}$, where $m \neq m'$. It implies that no FL worker is able to obtain a higher utility by changing its current coalition through a switch rule.
- *Individual-stable:* A partition $\Pi = \{S_1, \ldots, S_m, \ldots, S_M\}$ is individually-stable if there do not exist (i) an FL worker j in coalition S_m strictly prefers another coalition $S_{m'}$, where $m \neq m'$, i.e., $S_{m'} \bigcup \{j\} \succ_j S_m$, and (ii) the formation of a new coalition does not reduce the utilities of the members of the new coalition, i.e., $S_{m'} \bigcup \{j\} \succ_{j'} S_{m'}, \forall j' \in S_{m'}$.

Note that a partition that is Nash-stable is also individually-stable [40].

Proposition 1. *Given any initial partition Π, it always converges to a final partition $\Pi^* = \{S_1^*, \ldots, S_m^*, \ldots, S_M^*\}$ that is both Nash-stable and individually-stable.*

Proof. Given any current partition $\Pi_{curr} = \{S_1, \ldots, S_m, \ldots, S_M\}$, any FL worker $j \in \mathcal{J}$ is able to leave its current coalition and join another coalition by performing a switch operation. According to Definition 5, the utility of the FL worker is increased in its new coalition after performing the switch operation which results in a change in the partition. Based on (12) which defines the preference function of the FL worker, each switch operation results in a new partition that is not contained in its history set. Therefore, the hedonic coalition formation game consists of a sequence of switch operations, resulting in a sequence of partitions where each partition has not been visited before. Given any partition, the switch operations will eventually terminate and the hedonic coalition formation algorithm will converge to a final partition $\Pi^* = \{S_1^*, \ldots, S_m^*, \ldots, S_M^*\}$, where there is no incentive for any FL worker to change its current coalition. Given the final partition Π^*, the utility of each FL worker $j, \forall j \in \mathcal{J}$ is maximized.

The proposition is proved by contradiction. Suppose that the final partition Π^* is not Nash-stable. According to the definition of Nash-stability, there exist some FL workers that have incentive to leave its current coalition and join another coalition to increase their utilities by performing switch operations. Consequently, the partition Π^* is updated based on the switch rule in Definition 5. Thus, the partition Π^* is not final, which contradicts with our assumption that the final partition is not Nash-stable. Thus, the hedonic coalition formation algorithm always converges to a final partition Π^* that is Nash-stable. For a final partition that is Nash-stable, it is also individually-stable. □

4.3 Hedonic Coalition Formation Algorithm

To obtain the solution of the game, the hedonic coalition formation algorithm is proposed to enable the FL workers to choose the coalitions to join so that their utilities are maximized in the final partition. The algorithm for hedonic coalition formation among the FL workers is presented in Algorithm 1.

Algorithm 1. Algorithm for Hedonic Coalition Formation of FL Workers Using Switch Rule

Input: Set of FL workers, $\mathcal{J} = \{1, \ldots, j, \ldots, J\}$, set of cluster heads, $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$
Output: Final partition $\Pi^* = \{S_1^*, \ldots, S_m^*, \ldots, S_M^*\}$
1: $\Pi^* = \emptyset$
2: Initialize a partition Π_{curr} by randomly allocating J FL workers to M cluster heads
3: Initialize history set of all FL workers, i.e., $h(j) = S_m, \forall j \in \mathcal{J}$, where S_m is the coalition that FL worker j is allocated to
4: ***Switch Rule:***
5: **while** $\Pi_{curr} \neq \Pi^*$ **do**
6: Update the final partition such that $\Pi^* = \Pi_{curr}$
7: **for each** FL worker $j \in \mathcal{J}$ (worker j is in coalition $S_m \in$ partition Π_{curr}) **do**
8: Compute $x_j^{S_m}$
9: **for each** possible coalition $S_{m'} \in \Pi_{curr}$, $i \neq i'$ **do**
10: Compute $x_j^{S_{m'}}$
11: Compare $x_j^{S_m}$ and $x_j^{S_{m'}}$
12: **if** $x_j^{S_{m'}} > x_j^{S_m}$ **and** $S_{m'} \notin h(j)$ **then**
13: FL worker j leaves its current coalition, i.e., $S_m \leftarrow S_m \backslash \{j\}$
14: FL worker j joins a new coalition, i.e., $S_{m'} \leftarrow S_{m'} \bigcup \{j\}$
15: FL worker j updates its history set, $h(j)$ by adding the newly-joined coalition $S_{m'}$ into $h(j)$
16: Update the current partition Π_{curr}, i.e., $\Pi_{curr} \leftarrow \Pi_{curr} \backslash \{S_m, S_{m'}\} \bigcup \{S_m \{j\}, S_{m'} \bigcup \{j\}\}$
17: **end if**
18: **end for**
19: **end for**
20: **end while**
21: **return**Final partition $\Pi^* = \{S_1^*, \ldots, S_m^*, \ldots, S_M^*\}$ that is Nash-stable

Given that there are J FL workers, a partition Π_{curr} is first initialized by randomly allocating the FL workers to the M cluster heads (line 2). Besides, the history sets of all FL workers are initialized (line 3). The hedonic coalition formation game is based on the switch rule defined in Definition 5. Any FL worker $j \in \mathcal{J}$ is able to perform a switch operation to increase its utility. In particular, given a partition Π_{curr}, FL worker j leaves its current coalition S_m and join another coalition $S_{m'}$ where $m \neq m'$ and $S_{m'} \subseteq \Pi_{curr}$ if and only if it achieves higher utility by joining coalition $S_{m'}$ than that of current coalition S_m. The switch operation is illustrated from the perspective of an FL worker $j \in \mathcal{J}$. The FL worker j first computes its utility in the current coalition S_m, $x_j^{S_m}$ based on (10) (line 8). Given the current partition Π_{curr}, FL worker j evaluates other coalition $S_{m'}$, where $m \neq m'$ and $S_{m'} \subseteq \Pi_{curr}$, that it could possibly join (line 9-10). Specifically, it computes the utility $x_j^{S_{m'}}$ that it can achieve by joining coalition $S_{m'}$ using (10) (line 10). The FL worker j then compares the utilities of joining each of the coalitions based on the preference function defined in (12) (line 11). Hence, the preference order between the current coalition S_m and another coalition $S_{m'}$ is determined. If the utility of FL worker j is higher from joining coalition $S_{m'}$ than that of staying in the current coalition S_m and the coalition $S_{m'}$ has not been visited before, FL worker j performs a switch operation (lines 12-16). The FL worker leaves its current coalition S_m (line 13) and joins the new coalition $S_{m'}$ (line 14). The FL worker also updates its history set by adding the newly-joined coalition $S_{m'}$ into $h(j)$ (line 15). Given the changes in the coalitions, the current partition Π_{curr} is updated (line 16).

If the utility of FL worker j is higher for staying in its current coalition S_m than that of joining coalition $S_{m'}$ or the coalition $S_{m'}$ is found in its history set $h(j)$, the FL worker j does not leave its current coalition, and thus the current partition Π_{curr} remains unchanged. For the next iteration, the FL worker j considers to join other possible coalitions in current partition Π_{curr} by comparing the achievable utilities by using the preference function defined in (12). The process repeats until the algorithm converges to a Nash-stable partition. In particular, there is no FL worker j ($\forall j \in \mathcal{J}$), has the incentive to leave its current coalition and join another coalition given the current partition Π_{curr}. The switch mechanism terminates where there is no further change to the current partition Π_{curr}. At the end of the algorithm, the final partition $\Pi^* = \{S_1^*, \ldots, S_m^*, \ldots, S_M^*\}$ that maximizes the utilities of all FL workers is returned (line 21).

5 Simulation Results

In this section, we evaluate the performance of the hedonic coalition formation game in the SHFL network. We consider a network with 3 cluster heads and 10 FL workers. The simulation parameter values are summarized in Table 2.

Figs. 2 and 3 show the changes in utilities of FL workers and partitions after each switch operation respectively. A switch operation is performed when any FL worker is able to achieve a higher utility by leaving its current coalition and joining another coalition. For each given partition, each FL worker evaluates its utility by comparing the utility achieved in the current coalition against the possible utility gain from joining other coalitions based on the utility and preference function defined in (10) and (12) respectively. As a result, each worker may perform more than one switch operation. As an illustration of the hedonic coalition formation game algorithm, the FL workers are first randomly allocated to the cluster heads where FL workers 1, 3 and 5 are allocated to cluster head 1, FL workers 4, 6, 7, 8 and 10 are allocated to cluster head 2, whereas FL workers 2 and 9 are allocated to cluster head 3. Then, FL

TABLE 2
System Simulation Parameter Values

Parameter	Values
Fade parameter, z	(0,1)
Influence of uncertainty on reputation score, ω	[0,1]
Data quantity of FL worker j, d_j	[1000,10000]
Computational coefficient, μ [37]	10^{-26}
Total number of CPU Cycles, a_j	1GHz-3GHz
Computational capability of FL worker j, f_j	$[10^7, 10^8]$
Unit computation cost, θ_1	1
Transmission power of FL worker j, P_j	1W-5W
Channel Gain, $h_{j,m}$	5dBm-25dBm
Noise Spectral Density, N_0	-174dBm/Hz
Unit communication cost, θ_2	1
Reward pool of cluster head m, α_m	$[10^5, 10^6]$
Reputation score of cluster head m, γ_m	[100, 600]
Bandwidth of cluster head m, B_m	5MHz-20MHz
Congestion coefficient of cluster head m, β_m	(0,1)
Data size of local model parameters, W	100MB

worker 2 leaves cluster head 3 where it achieves a utility of 1930 and join cluster head 1 to gain a higher utility of 2469. The decision of FL worker 2 to join cluster head 1 causes the utility of FL worker 3 to decrease from 3755 to 2598. As such, the FL worker 3 leaves cluster head 1 to join cluster 2 to gain a utility of 2733. As there are too many FL workers that join cluster head 2, each FL worker is only able to gain a small share of the reward pool of cluster head 2. This results in the decision of FL worker 4 to leave cluster head 2 and join cluster head 1. FL worker 5 then leaves cluster head 1 and join cluster head 3 and so on. The hedonic coalition formation game algorithm converges to a final partition that is Nash-stable, i.e., FL workers 2, 4, 7, 8 and 10 join cluster head 1, FL workers 1, 3 and 6 join cluster head 2, whereas FL workers 5 and 9 join cluster head 3. Given this final partition, no FL worker is able to gain a higher utility by leaving its current coalition and joining another coalition. For example, in the final partition, FL worker 1 achieves a utility of 3303 by joining cluster head 2, but it is only able to achieve utilities of 2163 and 1429 for joining cluster head 1 and 3 respectively. Hence, it does not have incentive to leave its current coalition.

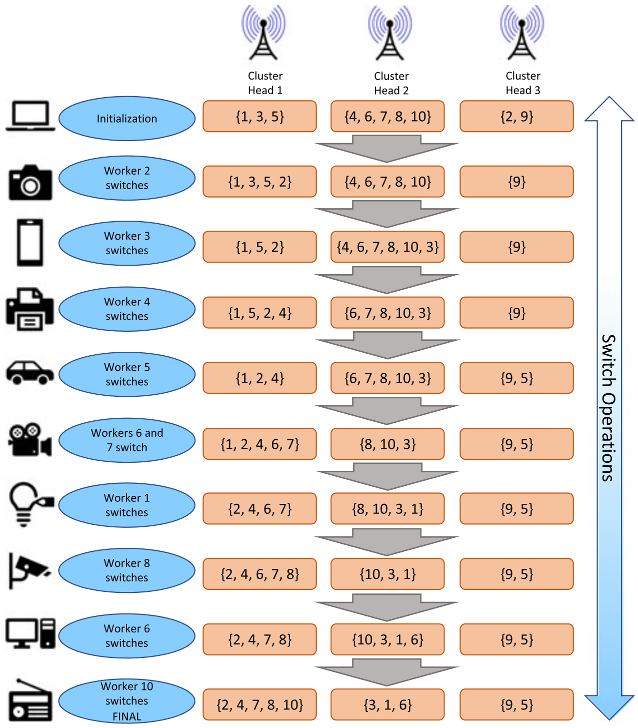
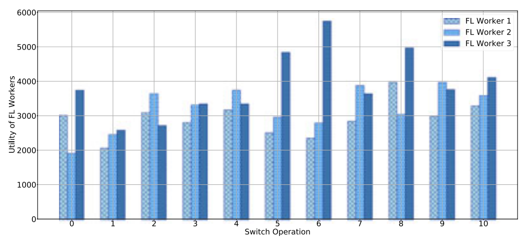
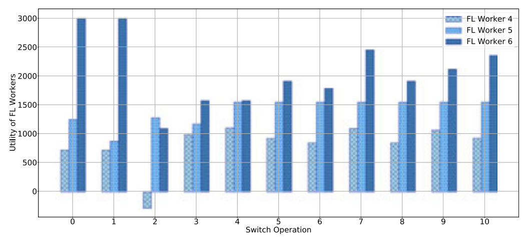


Fig. 3. Changes in partition by performing switch operations.

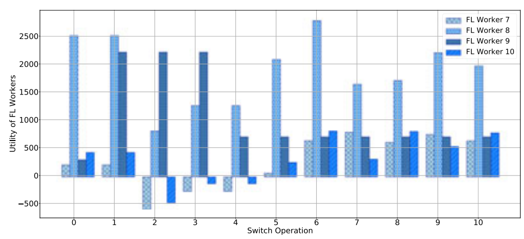
From Fig. 2 which show the changes in the utilities of FL workers after each switch operation, we observe that for each switch operation performed by an FL worker, the utility of that FL worker increases regardless the effect of its decision on other FL workers in the coalition, e.g., when FL worker 2 moves from cluster head 3 to cluster head 1 after the first switch operation, the utility of FL worker 2 increases from 1930 to 2469, but the utilities of FL workers 1, 3 and 5 decrease from 3025 to 2079, 3755 to 2598 and 1255 to



(a) Utilities of FL workers 1, 2 and 3



(b) Utilities of FL workers 4, 5 and 6



(c) Utilities of FL workers 7, 8, 9 and 10

Fig. 2. Utility of FL workers after each switch operation.

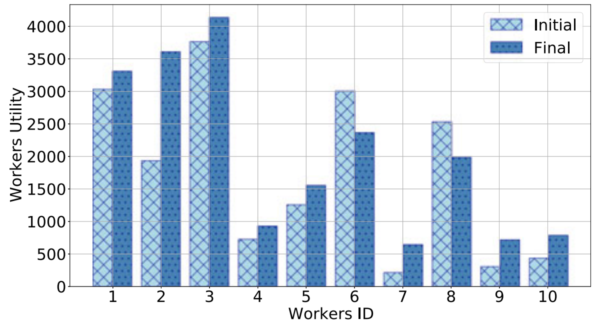


Fig. 4. Initial and final utilities of FL workers.

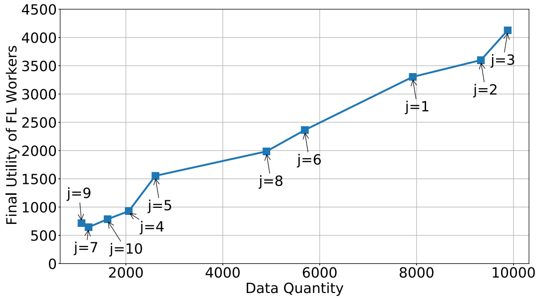


Fig. 5. Utility of FL workers versus data quantity.

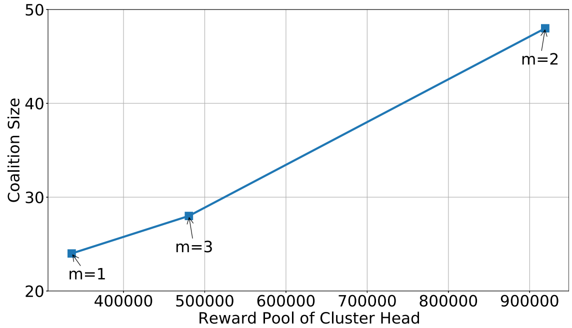


Fig. 6. Coalition size versus reward pool.

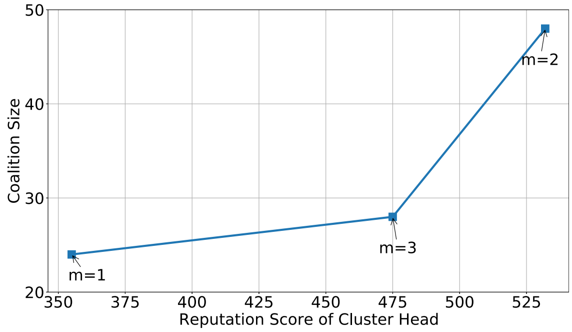


Fig. 7. Coalition size versus reputation score.

878 respectively. This reflects the selfish behaviour of the FL workers in maximizing their own utilities. Similarly, when FL worker 1 leaves cluster head 1 and joins cluster head 2 at the 7^{th} switch operation, its utility increases from 2371 to 2858, but the utilities of FL workers 3, 8 and 10 decrease from 5757 to 3654, 2796 to 1657 and 819 to 316 respectively.

We also compare the utilities of FL workers at the start and end of the hedonic coalition formation game algorithm in Fig. 4. Generally, the utilities of the FL workers increase. However, the utilities of some FL workers, e.g., FL workers 6 and 8, decrease. This again reflects the selfish behaviour of the FL workers where their decisions may cause harm to other FL workers in the coalitions. Furthermore, we study the effect of data quantity on the utilities of the FL workers. In Fig. 5, we observe that as the data quantity of FL workers increases, their utility also increases. In particular, FL worker 3 that has a larger data quantity of 9880, gains a utility of 4127 whereas FL worker 10 that has a smaller data quantity of 1622 only gains a utility of 786.

To study the hedonic coalition formation game in practical situations, we also consider a larger network with 3 cluster heads and 100 FL workers. Fig. 6 shows that as the amount of reward pool of the cluster head increases, the coalition size also increases. Cluster head 2 that offers the largest amount of reward pool of 919,186 has a coalition size of 48 whereas cluster head 1 that offers a reward pool of 336,205 only has 24 FL workers in its coalition. Fig. 7 shows the effect of the reputation score of the cluster heads on the coalition size. The cluster head with a higher reputation score, i.e., cluster head 2 with a reputation score of 532, has a larger coalition, i.e., 48 coalition members.

6 Conclusion

In this paper, we propose the Serverless Hierarchical Federated Learning (SHFL) framework that aims to achieve both communication latency reduction without the drawbacks of having a single point of failure. To improve the network efficiency, we introduce a reputation aware hedonic coalition formation scheme to model the cluster formation in the SHFL network. Then, we perform numerical simulations to validate the performance of the network. Simulation results show that the FL workers make decisions to maximize their own utilities, regardless of the effect of their decisions on other FL workers. The hedonic coalition formation algorithm converges to a Nash-stable partition. For the future works, we can consider the use of a blockchain to store the reputation scores in the network.

References

[1] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, “Internet of medical Things: A review of recent contributions dealing with cyber-physical systems in medicine,” *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3810–3822, Oct. 2018.
[2] W. Y. B. Lim *et al.*, “When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 1, pp. 457–466, Jan. 2022.
[3] D. Wood, N. Apthorpe, and N. Feamster, “Cleartext data transmissions in consumer IoT medical devices,” in *Proc. Workshop Internet Things Secur. Privacy*, 2017, pp. 7–12.
[4] W. Y. B. Lim *et al.*, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Jul./Sep. 2020.
[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.
[6] A. Hard *et al.*, “Federated learning for mobile keyboard prediction,” 2019, *arXiv: 1811.03604*.
[7] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, “Federated learning for healthcare informatics,” *J. Healthcare Inform. Res.*, vol. 5, no. 1, pp. 1–19, 2021.
[8] J. Pang, Y. Huang, Z. Xie, J. Li, and Z. Cai, “Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution,” *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 759–771, 2021.

[9] Y. Liu *et al.*, "Federated learning in the sky: Aerial-ground air quality sensing framework with UAV swarms," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021.

[10] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Gener. Comput. Syst.*, vol. 117, pp. 328–337, 2021.

[11] M. S. H. Abad, E. Ozfatura, D. GUndUz, and O. Ercetin, "Hierarchical federated learning ACROSS heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 8866–8870.

[12] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.

[13] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[14] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.

[15] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 8851–8855.

[16] J. Konen, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[17] N. Yoshida, T. Nishio, M. Morikura, and K. Yamamoto, "MAB-based client selection for federated learning with uncertain resources in mobile networks," in *Proc. IEEE Globecom Workshops*, 2020, pp. 1–6.

[18] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Robust federated learning with noisy communication," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3452–3464, Jun. 2020.

[19] J. S. Ng *et al.*, "Joint auction-coalition formation framework for communication-efficient federated learning in UAV-enabled internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2326–2344, Apr. 2021.

[20] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[21] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[22] L. U. Khan, M. Alsenwi, Z. Han, and C. S. Hong, "Self organizing federated learning over wireless networks: A socially aware clustering approach," in *Proc. Int. Conf. Inf. Netw.*, 2020, pp. 453–458.

[23] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.

[24] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 634–643.

[25] Y. Khazbak, J. Fan, S. Zhu, and G. Cao, "Preserving personalized location privacy in ride-hailing service," *Tsinghua Sci. Technol.*, vol. 25, no. 6, pp. 743–757, 2020.

[26] W. Y. B. Lim *et al.*, "Hierarchical incentive mechanism design for federated machine learning in mobile networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9575–9588, Oct. 2020.

[27] 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.

[28] K. Xu, F. Wang, H. Wang, and B. Yang, "Detecting fake news over online social media via domain reputations and content understanding," *Tsinghua Sci. Technol.*, vol. 25, no. 1, pp. 20–27, 2020.

[29] W. Y. B. Lim *et al.*, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022.

[30] L. Ruan, J. Chen, Q. Guo, H. Jiang, Y. Zhang, and D. Liu, "A coalition formation game approach for efficient cooperative multi-UAV deployment," *Appl. Sci.*, vol. 8, no. 12, 2018, Art. no. 2427.

[31] F. Afghah, M. Zaeri-Amirani, A. Razi, J. Chakareski, and E. Bentley, "A coalition formation approach to coordinated task allocation in heterogeneous UAV networks," in *Proc. Ann. Amer. Control Conf.*, 2018, pp. 5968–5975.

[32] T. Mavlanova, R. Benbunan-Fich, and M. Koufaris, "Signaling theory and information asymmetry in online commerce," *Inf. Manage.*, vol. 49, no. 5, pp. 240–247, 2012.

[33] M. Sohail, L. Wang, S. Jiang, S. Zaineldeen, and R. U. Ashraf, "Multihop interpersonal trust assessment in vehicular Ad-hoc networks using three-valued subjective logic," *IET Inf. Secur.*, vol. 13, no. 3, pp. 223–230, Apr. 2019.

[34] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1389–1399, Jun. 2018.

[35] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2020, *arXiv: 1808.04866*.

[36] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Performance optimization of federated learning over mobile wireless networks," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2020, pp. 1–5.

[37] J. Zhang *et al.*, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[38] X. Gong, L. Duan, X. Chen, and J. Zhang, "When social network effect meets congestion effect in wireless networks: Data usage equilibrium and optimal pricing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 449–462, Feb. 2017.

[39] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, "Hedonic coalition formation for distributed task allocation among wireless agents," *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1327–1344, Sep. 2011.

[40] A. Bogomolnaia and M. O. Jackson, "The stability of hedonic coalition structures," *Games Econ. Behav.*, vol. 38, no. 2, pp. 201–230, 2002.

Jer Shyuan Ng received the graduation (Double Hons.) degree in electrical engineering and economics from the National University of Singapore in 2019. She is currently working toward the Alibaba PhD degree with the Alibaba Group and Alibaba-NTU Joint Research Institute, Nanyang Technological University (NTU), Singapore. Her research interests include incentive mechanisms and edge computing.

Wei Yang Bryan Lim is currently working toward the Alibaba Talent Programme PhD degree with the Alibaba-NTU Joint Research Institute (JRI), Nanyang Technological University (NTU), Singapore. He was a reviewer in leading journals and flagship conferences and is currently the assistant to the editor-in-chief of the *IEEE Communications Surveys & Tutorials*.

Zehui Xiong (Member, IEEE) received the PhD degree from Nanyang Technological University, Singapore. He is currently an assistant professor with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design. He was a researcher with Alibaba-NTU Joint Research Institute, Singapore. He was the visiting scholar with Princeton University and the University of Waterloo. He has authored or coauthored more than 120 research papers in leading journals and flagship conferences and many of them are ESI Highly Cited Papers. His research interests include wireless communications, network games and economics, blockchain, and edge intelligence. He is the editor or the guest editor for many leading journals including the IEEE *Journal on Selected Areas in Communications*, *Transactions on Vehicular Technology*, *Internet of Things Journal*, *Transactions on Cognitive Communications and Networking*, *Transactions on Network Science and Engineering*, and *IEEE Systems Journal*. He was the recipient of numerous best paper awards in international conferences, IEEE TCSC Early Career Researcher Award for Excellence in Scalable Computing, IEEE SPCC Technical Committee Best Paper Award, IEEE VTS Singapore Best Paper Award, Chinese Government Award for Outstanding Students Abroad, and NTU SCSE Best PhD Thesis Runner-Up Award. He is the founding vice chair of Special Interest Group on Wireless Blockchain Networks in IEEE Cognitive Networks Technical Committee.

Xianbin Cao (Senior Member, IEEE) is currently the dean and a professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His research interests include intelligent transportation systems, airspace transportation management, and intelligent computation.

Jiangming Jin (Member, IEEE) received the bachelor's degree from the University of Electronic Science and Technology of China in 2008, the graduation degree from Singapore Nanyang Technological University in 2013, and the PhD degree in computer engineering. He started his career with J. P. Morgan (Singapore and Beijing). In J. P. Morgan, he was with the most sophisticated and large-scale financial computing system and also with JPM Credit Portfolio Group in credit derivative calculations. He began a venture journey in autonomous driving with TuSimple (NASDAQ: TSP) in 2017. In TuSimple, he was successively as the director of HPC and a senior expert of engineering to oversee projects of high performance robotic middleware and heterogeneous computing systems. He has authored or coauthored more than 20 research papers in top conferences or journals and applied more than ten U.S. or CN patents. His research interests include fintech, AI chips and software, big data and machine learning systems, 5G, and IoT. He was a session-chair or a keynote-speaker in several AI Summits or Forums. He is an executive member of CCF-TCHPC and CCF-TCARCH. He holds several advisory positions with Universities and Non-Profit Institutes, including Shanghai Tech University, Tianjin University, and Shenzhen Fintech Association.

Dusit Niyato (Fellow, IEEE) received the BE degree from King Mongkuk's Institute of Technology Ladkrabang (KMITL), Thailand, in 1999 and the PhD degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a professor with the School of Computer Science and Engineering and, by courtesy, School of Physical and Mathematical Sciences, the Nanyang Technological University, Singapore. He has authored or coauthored more than 380 technical papers in the area of wireless and mobile networking, and is an inventor of four U.S. and German patents. He has authored four books including a *"Game Theory in Wireless and Communication Networks: Theory, Models, and Applications"* with Cambridge University Press. He was the recipient of Best Young Researcher Award of IEEE Communications Society (ComSoc) Asia Pacific (AP) and the 2011 IEEE Communications Society Fred W. Ellersick Prize Paper Award. He is currently the senior editor of the *IEEE Wireless Communications Letter*, the area editor of *IEEE Transactions on Wireless Communications* (Radio Management and Multiple Access), the area editor of *IEEE Communications Surveys and Tutorials* (Network and Service Management and Green Communication), the editor of *IEEE Transactions on Communications*, an associate editor of the *IEEE Transactions on Mobile Computing*, the *IEEE Transactions on Vehicular Technology*, and the *IEEE Transactions on Cognitive Communications and Networking*. He was the guest editor of *IEEE Journal on Selected Areas on Communications*. He was a distinguished lecturer of the IEEE Communications Society for 2016–2017. He was named the 2017, 2018, and 2019 highly cited researcher in computer science.

Cyril Leung received the BSc (First Class Hons.) degree from Imperial College, University of London, U.K., and the MS and PhD degrees in electrical engineering from Stanford University. He has been an assistant professor with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, and the Department of Systems Engineering and Computing Science, Carleton University. Since 1980, he has been with the Department of Electrical and Computer Engineering, University of British Columbia (UBC), Vancouver, Canada, where he is currently a professor and currently holds the PMC-Sierra professorship in Networking and Communications. From 2008 to 2011, he was an associate dean of research and graduate studies with the faculty of Applied Science, UBC. His research interests include wireless communication systems, data security, and technologies to support ageless aging for the elderly. He is a member of the association of professional engineers and geoscientists of British Columbia, Canada.

Chunyan Miao received the BS degree from Shandong University, Jinan, China, in 1988, and the MS and PhD degrees from Nanyang Technological University, Singapore, in 1998 and 2003, respectively. She is currently a professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU) and the director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY). Her research interests include infusing intelligent agents into interactive new media (virtual, mixed, mobile, and pervasive media) to create novel experiences and dimensions in game design, interactive narrative, and other real world agent systems.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.