IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 13 November 2020; revised 7 January 2021; accepted 23 February 2021.
Date of publication 3 March 2021; date of current version 7 June 2022.

Digital Object Identifier 10.1109/TETC.2021.3063517

A Survey of Incentive Mechanism Design for Federated Learning

YUFENG ZHAN, JIE ZHANG, ZICONG HONG, LEIJIE WU, PENG LI, (Senior Member, IEEE), AND SONG GUO, (Fellow, IEEE)

Yufeng Zhan is with the School of Automation, Beijing Institute of Technology, Beijing 100811, China.
Jie Zhang, Zicong Hong, Leijie Wu, and Song Guo are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, China.
Peng Li School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu 965-8580, Japan.
CORRESPONDING AUTHOR: PENG LI (pengli@u-aizu.ac.jp)

ABSTRACT Federated learning is promising in enabling large-scale machine learning by massive clients without exposing their raw data. It can not only enable the clients to preserve the privacy information, but also achieve high learning performance. Existing works of federated learning mainly focus on improving learning performance in terms of model accuracy and learning task completion time. However, in practice, clients are reluctant to participate in the learning process without receiving compensation. Therefore, how to effectively motivate the clients to actively and reliably participate in federated learning is paramount. As compared to the current incentive mechanism design in other fields, such as crowdsourcing, cloud computing, smart grid, etc., the incentive mechanism for federated learning is more challenging. First, it is hard to evaluate the training data value of each client. Second, it is difficult to model the learning performance of different federated learning algorithms. In this article, we survey the incentive mechanism design for federated learning. In particular, we present a taxonomy of existing incentive mechanisms for federated learning, which are subsequently discussed in depth by comparing and contrasting different approaches. Finally, some future directions of how to incentivize clients in federated learning have been discussed.

INDEX TERMS Federated learning, incentive mechanism, survey

I. INTRODUCTION

Machine learning [1] based on artificial neural networks has achieved unprecedented success in various fields, e.g., computer vision, robotics, human-machine games, etc. In order to obtain a better machine learning model, large amounts of training data should be fed to the machine learning model. However, most of current training data are generated by resource-constrained devices (e.g., tablets, smartphones, etc.). Due to the limited communication bandwidth and privacy concerns, uploading such large amounts of data to the cloud for centralized model training is often impractical.

Thanks to the rapid development of edge computing [2], [3] and distributed machine learning [4], a new paradigm of distributed machine learning named federated learning [5] has been proposed to let distributed edge devices train machine learning models cooperatively without exposing their raw data. Federated learning usually adopts the parameter-server architecture, where clients train local models that are synchronized by a parameter server. A typical federated learning process contains multiple rounds. In each round, clients download the new machine learning model from the parameter server and train the local models with their own data by multiple epochs, respectively. They then upload the newly updated models to a server residing in the remote cloud that creates a new global machine learning model. Once federated learning is proposed [6], it attracts widespread attentions from academia [7]–[11] and industry [12]. Wang *et al.* [7] propose an adaptive local epochs control in each training round due to the clients are resource-constrained. Since the federated learning systems are easy to be attacked by the malicious clients, a new model aggregation approach [10] has been proposed to resist the Byzantine attacks. Besides, the non-identically distributed (Non-IID) data across the clients is another key challenge in federated learning, it decreases the efficiency of SGD-based training approach. To tackle the Non-IID challenge, model averaging and data sharing approaches have been proposed. McMahan *et al.* [6] present the FedAVG approach for the federated learning based on iterative model averaging.

However, the model accuracy of FedAVG decreases significantly, by up to 11 percent for MNIST, 51 percent for CIFAR-10 and 55 percent for keyword spotting datasets, with highly skewed Non-IID data. Exiting algorithms for federated learning mainly use gradient descent for loss function optimization. However, by taking the preceding results into current gradient update can potentially accelerate the convergence. In [13], Liu *et al.* propose the momentum federated learning algorithm using momentum gradient descent in the local step update which can significantly accelerate the convergence of federated learning.

The above federated learning systems heavily rely on the quality of clients' local model updates. However, clients may reluctant to participate and share their model updates without sufficient compensation. Indeed, participating in the federated learning tasks incurs system costs. For example, when a client participates in a federated learning task, it is inevitable to consume resources of his/her devices, including computation, communication, and battery power. In addition, the federated learning framework still faces various security risks. For example, Song *et al.* [14] show that the important information of training data can be inferred by the intermediate gradients. Besides, a curious parameter server can learn the private information of clients' training data through generative adversarial networks. Because of these risks, clients are more reluctant to take part in federated learning tasks, unless they can get sufficient rewards.

Therefore, rewards need to be introduced into the federated learning systems because: (i) participating in federated learning will incur computational resources consumption, network bandwidth usage, and shortened battery life for the clients, and sufficient rewards can motivate them to tolerate these costs and make contributions; and (ii) unlike cloud-based distributed machine learning where the parameter servers can control all workers' behaviors, the workers in federated learning are independent, and only their owners can determine when, where and how to participate in federated learning. Therefore, rewards can be used to somehow affect the clients' decisions. With different incentive mechanisms, the clients will execute different training strategies, and thus affect the final machine learning model performance. Within a federated learning system, there are two main challenges: (i) how to evaluate each client's contribution, and (ii) how to recruit and retain more clients. The first challenge is from the parameter server's perspective, as different learning tasks need clients to train different machine learning models on different training data, so how to obtain higher learning performance by providing the minimum rewards is challenging. The second challenge comes from the clients' perspective, i.e., we cannot fully measure the effectiveness of the incentive mechanism designed for federated learning unless the clients' concerns, needs and goals are all met. That is, how to provide fair, profitable and secure learning opportunities to get enough clients participation.

We survey the related works of incentive mechanism design for federated learning during 2017-2020. We present state-of-the-art research efforts on incentive mechanism design driven by clients' contribution, reputation and resource allocation. The rest of this paper is organized as follows. We start from the background of this survey in Section II, how federated learning and incentive mechanism work are introduced. Second, different incentive mechanisms driven by clients' contribution are organized in Section III. Third, the existing works for incentive mechanism design driven by the clients' reputation are presented in Section IV. Last, we discuss how to incentivize the clients to allocate more resource in the federated learning in Section V. Finally, we discuss some possible future directions in Sections VI and VII draws the conclusions.

II. BACKGROUND

In this section, we briefly introduce some necessary backgrounds of federated learning and incentive mechanisms.

A. FEDERATED LEARNING

In recent years, with the development of artificial intelligence technologies and their widespread applications, data privacy preservation has been paid more and more attention. The process of data collection must be open and transparent among companies and institutions, data belonging to one user or company cannot be exchanged with others without user authorization. This leads to the massive data in the form of "data island", which lacks of effective communication and cooperation among the users and makes the successful implementation of artificial intelligence encounter difficulties.

The concept of federated learning was proposed to allow multiple clients to collaboratively train a shared model by iteratively aggregating model updates without exposing their raw data. The eligible clients first register on the parameter server. Then, the parameter server proceeds federated learning synchronously in rounds. At the beginning of each communication round, the parameter server first distributes the latest global model to the selected clients. Then the clients train the model on their local datasets, and upload the local model updates to the parameter server for aggregation.

The typical architecture and workflow of training process in a federated learning system can be illustrated in Figure 1. Generally, it contains two main components in the federated learning system, i.e., the clients/participants (the data owners) and the parameter server. We define $\mathcal{N} = \{1, \ldots, N\}$ as the set of N clients and the private dataset on client i is represented as $\mathcal{D}_i$. The data sample in $\mathcal{D}_i$ can be denoted by (x_j, y_j), where x_j is the the input/feature vector for sample $j \in \mathcal{D}_i$ and y_j is the corresponding label. Each client will train a local model with parameters $\boldsymbol{\omega}_i$ by using its own dataset $\mathcal{D}_i$ and send the local model parameters rather than the raw data to the parameter server. After receiving the local model parameters from all clients, the parameter server aggregates them to create a new global model. In this case, the centralized server does not need to collect all the local data into a whole dataset like $\mathcal{D} = \cup_{i \in \mathcal{N}} D_i$, which ensures the privacy. Let $f(x_j, y_j; \boldsymbol{\omega})$, or $f_j(\boldsymbol{\omega})$ for convenient, denote the loss

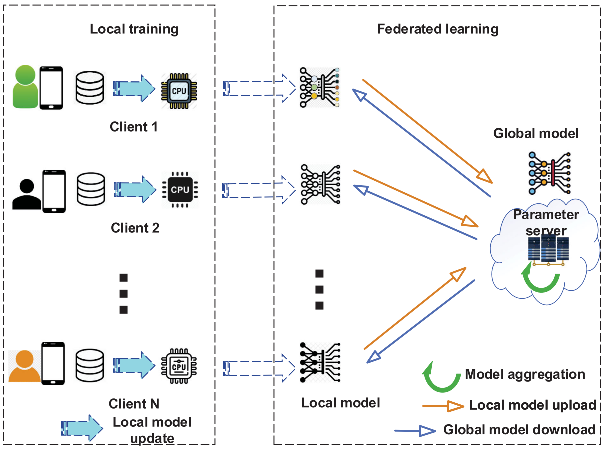


FIGURE 1. **An illustration of federated learning.**

function of sample j. In this framework, the data owners serve as the federated learning participants which collaboratively train a machine learning model. More specifically, the federated learning training process can be summarized into three steps: initialization, local model training, global aggregation.

Step 1: Initialization. The parameter server first decides the architecture of the global model, and initializes the parameters of the global model randomly or by pretraining on public dataset according to the training task. Then, the parameter server distributes the initial global model parameters ω_0 to the selected clients.

Step 2: Local model training: In the tth communication round, each selected client updates the local model ω_t^i based on the received global model parameters ω_t using their local dataset. After that, the updated local model parameters ω_{t+1}^i will be sent to the parameter server. The goal of client i in tth round is to minimize the empirical loss $F(\omega_t^i)$ based on the local dataset, i.e.,

$$\omega_t^i = \arg\min_{\omega_t^i} F\left(\omega_t^i\right), \tag{1}$$

$$F(\omega_t^i) = \frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i} f_j(\omega_t^i), \tag{2}$$

where $|\mathcal{D}_i|$ denotes the number of samples in dataset $\mathcal{D}_i$.

The update process in each client can be achieved by performing stochastic gradient descent (SGD) with min-batches sampled from its local dataset

$$\omega_t^i = \omega_t^i - \eta\nabla F(\omega_t^i), \tag{3}$$

where $\nabla F(\omega_t^i)$ represents the gradient of loss function, η is the learning rate.

Step 3: Global Aggregation: In each round, the parameter server aggregates the local updated parameters from selected clients, and replaces the global model by the average model. The updated global model parameters ω_{t+1} are then sent back to the selected clients. In particular, the aim is to minimize the global loss function, which can be expressed as follows:

$$F(\omega_t) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{N}|\mathcal{D}_i|F(\omega_t^i), i \in 1, 2, \ldots, N. \tag{4}$$

The above steps will be repeated until a desired accuracy is achieved. As compared to traditional model training approaches, federated learning has the following advantages:

- *Privacy preservation:* Since the raw data of clients does not need to be sent to the server, the privacy of clients can be guaranteed. In the long term, the guaranteed privacy can attract more participants to join the collaborative model training process, which further improves the performance of the machine learning model.
- *Efficient resource utilization:* In federated training, only the updates are required to be transmitted to the parameter server, which reduces the total communication overhead. Besides, by increasing computation on each client, where each client performs local training with more epochs between each training round, the communication resources can be used efficiently.
- *Lower inference latency:* With federated learning, each client can consistently train and update a machine learning model locally. The updated model can be used to make predictions on the client's own device. Compared with traditional method that making decision in the centralized server, local decision has lower latency.

B. INCENTIVE MECHANISM

The definition of incentive is something that motivates an individual to do some specific actions. The incentive is the core of all economic activities, both in terms of individual decision making and in terms of cooperation and competition within a larger institutional structure. Incentives can be classified into positive and negative. Positive incentives seek to motivate others by promising a reward, whereas negative incentives aim to avoid malicious behaviors by punishing the individuals.

Using game theory to design incentive mechanism has been widely studied in other areas, such as crowdsensing [15], edge computing [16], and etc. Yang *et al.* [15] have studied the platform-centric and user-centric crowdsourcing using Stackelberg game and auction approach, respectively. Authors in [17] propose the incentive mechanisms to motivate users to use device-to-device communication. They consider two different markets, one is open markets where users have information of all users and the other is sealed markets where users only have their own information. In [18], Zhan *et al.* show the effectiveness of incentive mechanism for opportunistic networks, and design the online and offline approaches, respectively. As the incentive mechanism design in above mentioned works, each participant's utility model can be built precisely, then game theory is applied to analyze each participant's behavior. In federated learning, how to quantify the value of each client's training data is

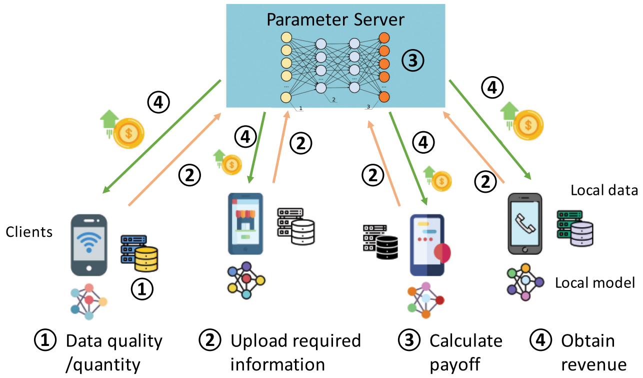


FIGURE 2. **An illustration of incentive mechanism for federated learning driven by clients' contribution.**

challenging. Meanwhile, due to the complexity of the federated learning algorithm, it is challenging to model the final learning performance of federated learning system. Hence, it is hard to model the utility function of each participant (parameter server and clients) in federated learning, which makes the existing works about incentive mechanism design cannot be directly applied.

III. INCENTIVE MECHANISM DESIGN DRIVEN BY CLIENTS' DATA CONTRIBUTION

In federated learning, many data owners may not actively participate in training a shared model, especially when the data owners are enterprises rather than individuals. As shown in Figure 2, participants tend to gain some revenue when take part in the federated learning with their local datasets. Therefore, it is essential to evaluate the contributions of different data providers so that the profits gained by the learning system can be distributed appropriately. A reasonable contribution evaluation criteria can further make the incentive mechanism attract more participants to join. A number of literature focuses on designing incentive mechanisms for federated learning by clients' contribution, which can be summarized into two categories: data quality and data quantity. Table 1 lists the related works in this topic.

A. DATA QUALITY

How to assess the data value is an increasingly common question posed by both organization and individual. Shapley value has been widely used in this area [30]. Sim *et al.* [31] study how to value each client's training data and design an incentive-aware reward scheme based on Shapley value to give each client a customized machine learning model as a reward instead of monetary compensations. Though the incentive mechanism for machine learning market has deeply studied how to value each client's training data, yet it does not take the training cost into consideration, which cannot be applied to federated learning directly.

In federated learning, to attract clients with high-quality data, Song *et al.* [19] propose an efficient and effective metric called contribution index based on Shapley value to evaluate the contribution of different clients in federated learning. In order to calculate the contribution index of different clients, the machine learning models with different combinations of training datasets are required to be trained and evaluated. Therefore, it will consume lots of time and energy, which is practically impossible. To overcome this problem, the authors reconstruct the models approximately on different combinations of training datasets through the intermediate results of federated learning so as to avoid extra training.

The auction mechanism has also been applied in federated learning. Since there exists a widening resource gap between different clients, and this gap might deteriorate the performance of federated learning seriously. Zeng *et al.* [22] consider the multi-dimensional and dynamic edge resources in federated learning, and present a novel multi-dimensional incentive framework for federated learning. They use the game theory to derive the optimal strategies for each clients, and leverage the expected utility to guild the parameter server to select the optimal clients to train the machine learning model.

In mobile crowdsensing, a client may have insufficient training data to build an efficient machine learning model. Authors in [21] propose a federated learning-based privacy-preserving approach to facilitate cooperative machine learning among clients. There are two main challenges of incentive mismatches between clients and users (which collect the training data), as well as among clients. The authors propose a hierarchical incentive mechanism architecture in multiple clients scenario. They use the contract theory to build the incentive mechanism between clients and users, and the

TABLE 1. **Summary of incentive mechanism design driven by clients' contribution.**

Approach	Literature	Applications	Models	Description
Data Quality	[19]	Classification	Any	Define the Contribution Index (CI) to measure the contribution of different clients
	[20]	—	Any	Data owner dynamically receive payoff according to their contributions
	[21]	Crowdsourcing	Any	Use the coalitional game theory approach based on marginal contributions
	[22]	Classification	CNN, LSTM	Auction theory
Data Quantity	[23]	Crowdsourcing	LeNet-5	Based on clients' sensing and training capabilities
	[24], [25]	IoT	Any	DRL-based reward allocation
	[26], [27]	Classification	CNN	Use Blockchain to provide auditability and fairness
	[28], [29]	—	Any	Multi-dimensional contract-theoretic approach

coalitional game theory are used to reward the clients based on their marginal contributions. With the backward induction, the contract formulation is solved first and then the conditional game.

In federated learning, the training and commercialization of the machine learning models will take time. Hence, there will be delays before the parameter server could pay back the clients. This mismatch between rewards and clients' contribution has not been studied in the aforementioned works. In order to achieve the long-term system performance and attract more clients with high-quality data, Yu *et al.* propose a fair incentive scheme, Federated Learning Incentivizer (FLI), to avoid the unfairness treatment during the training process of federated learning [20]. FLI can dynamically adjust clients' contributions based on three criteria in order to match the contributions with rewards. Theoretical analysis and extensive evaluations show that FLI can produce near-optimal utility and minimize the inequality among clients.

B. DATA QUANTITY

Since the accuracy of the learning model is related to the size of training samples [23], [35], several works adopt the training data quantity to measure the clients' contributions.

Game theory is a powerful tool to analyze the incentive of multiple participants and their behaviors in federated learning. Zhan *et al.* propose a game-based incentive mechanism for a federated learning platform which combines distributed deep learning and crowdsensing together for big data analytics on mobile clients [23]. The platform first publishes a task and issues the corresponding reward. In order to maximize its own utility, each mobile client decides its participation level, i.e., the quantity of training data, by taking its obtained reward and energy cost into consideration. The mobile clients' decision making problems are formulated as a non-cooperative game to achieve a Nash Equilibrium. Furthermore, by integrating game theory and deep reinforcement learning (DRL), Zhan *et al.* propose an DRL-based incentive mechanism for federated learning [24], [25]. To motivate clients to contribute model training, this work formulates the interactions between parameter server and clients as a Stackelberg game. In the game, the parameter server publishes the training task and announces a total reward as a leader and then clients decide the collected data quantity as followers. A DRL-based incentive mechanism is proposed to achieve the equilibrium under a privacy-protected and dynamic environment. Specifically, the parameter server acts as an DRL agent to decide an optimal payment, without need to accurately evaluate each client's contributions and obtain their private information in advance.

Unlike the aforementioned non-cooperative game theory based approaches [23]–[25], Ding *et al.* propose a multi-dimensional contract-theoretic approach to design the parameter server's optimal incentive mechanism in the presence of clients' multi-dimensional private information including training cost and communication delay [28], [29]. They perform the analysis in three different information scenarios,

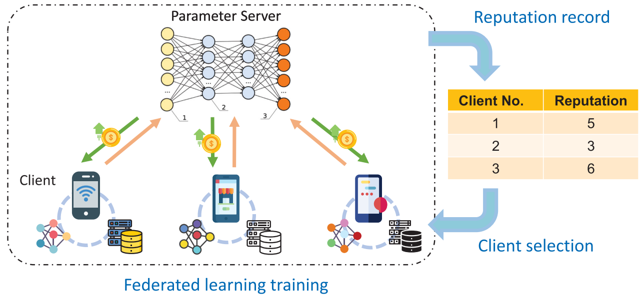


FIGURE 3. **An illustration of incentive mechanism design for federated learning driven by clients' reputation.**

include complete information, weakly incomplete information, and strongly incomplete information, to reveal the impact of information asymmetry on parameter server's incentive mechanism design.

To guarantee the security of incentive mechanism, some federated learning works adopt blockchain technology to record the training of clients and reward them with cryptocurrency. For example, Weng *et al.* [26] design a collaborative training framework in which clients jointly participate in deep learning model training. The sharing of local gradients for each client is in the form of transactions in the blockchain. Then, the workers collect and validate the transactions, each of which consists of a local gradient, and pack them into blocks. An incentive mechanism with compatibility and liveness properties is proposed to reward clients according to their processed data quantities and honest behaviors. Federated learning could fail if multiple clients abort training or send malformed parameters to the parameter server. Such misbehavior is not auditable and parameter server may compute incorrectly due to single point failure. Bao *et al.* [27] propose FLChain to bulid a decentralized, public auditable and healthy federated learning ecosystem with trust and incentive. In FLChain, honest client can gain fairly partitioned profit via a well-trained model according to its contribution and the malicious clients can be timely detected and heavily punished.

IV. INCENTIVE MECHANISM DESIGN DRIVEN BY CLIENTS' REPUTATION

Reputation is an important metric for client selection during federated learning. Client with higher reputation have a higher probability to bring high-quality and reliable training for federated learning tasks. As shown in Figure 3, at the end of each training task, the reputation of clients is updated based on their behaviors then the client selection in the next training takes the reputation records into consideration. Table 2 concludes the related works in this topic.

Zhao *et al.* present a blockchain-based reputation system for the federated learning of home appliance manufacturers to train a machine learning model based on customers' data [32]. At the beginning, each customer has an equal reputation value recorded in the blockchain. The reputation of a customer increases when he contributes correct and useful model

TABLE 2. Summary of incentive mechanism design driven by clients' reputation.

Literature	Applications	Models	Storage	Description
[32]	Crowdsourcing	CNN		Customers have the reputation records
[33]	—	Any	Blockchain	Edge devices, fog nodes, and cloud servers have the reputation records
[34]	—	Any		The reputations of clients are given according to poisoning attack detection schemes and multiweight subjective logic model

parameters and decreases when he uploads malicious model parameters. For a customer with a higher reputation, his opportunities to be selected in the next training task are higher.

Unlike [32] in which only customers (clients) have reputation records, Rehman *et al.* present a blockchain-based reputation system in which all three kinds of participants, i.e., the edge devices, fog nodes, and cloud servers, can grade each other [33]. Specifically, edge devices can grade fog nodes and cloud servers after requesting model parameters from them. Likewise, fog nodes can grade the connected edge devices according to their data-richness, context-awareness, etc. The cloud servers can grade fog nodes and edge devices according to their participation-activeness, sharing-willingness, etc. The system aggregates, calculates and records the reputation of each participant in federated learning via smart contract. Depending on the reputation-aware incentive mechanism, the participants can be benefited by the honest and high quality behaviors, which promotes healthy development of federated learning.

The above two works have two shortages. First, the scoring mechanism is too subjective and lacks quality evaluation schemes. Second, each participant only has a score and can be easily affected by malicious raters. Kang *et al.* present a blockchain-based reputation system for the reliable federated learning [34]. A task publisher detects the attackers and unreliable clients using RONI scheme for IID scenarios and FoolsGold scheme for non-IID scenarios. Based on the detecting results, the publisher can update the reputation of its interacted clients. For each client, to combine and relate different scores given by all task publishers, its composite reputation value will be generated by a multiweight subjective logic model.

V. INCENTIVE MECHANISM DESIGN DRIVEN BY CLIENTS' RESOURCE ALLOCATION

Federated learning is a distributed machine learning framework that usually consists of a parameter server and a crowd of various clients, e.g., personal computers, smartphones, wearable devices, etc. The heterogeneity among these devices makes the federated learning training process subject to different resource constraints including bandwidth, storage, and energy. Therefore, as shown in Figure 4, how to design an incentive mechanism for resource allocation among different clients becomes critical. In the following parts, we focus on two types of resource allocation for federated learning: computational resource and communication resource. The related works are listed in Table 3.

A. COMPUTATIONAL RESOURCE

In terms of computational resources, Sarikaya *et al.* first analyze the influence of heterogeneous clients on federated learning convergence, based on that, they propose an incentive mechanism to balance the time delay of each iteration [36]. More specifically, the parameter server has a limited budget and distributes its budget among clients to motivate them to contribute their CPU power and achieves a fast convergence with a target accuracy. They present a Stackelberg game to improve its performance by optimizing the computational resource allocation strategy among clients as well as the budget allocation of the parameter server. Khan *et al.* [37] also formulate the incentive mechanism using the Stackelberg game to motivate clients to participate in the federated learning. Nevertheless, they pay more attention to enable federated learning at network edge.

Different from the above game-based methods, Zhan *et al.* propose an DRL-based approach to design the incentive mechanism and find the optimal trade-off between model training time and parameter server's payment under a dynamic network environment [39]. The traditional theoretical analysis methods are inapplicable here with hardness of the federated learning algorithm and the dynamic learning environment, which makes it difficult to seek optimal solution. On the contrary, the DRL approach can improve its current strategy through accumulated experience gained during the previous training process, so empirically bring the current strategy close to the optimal solution.

B. COMMUNICATION RESOURCE

There are some other works focus on communication resource allocation. Le *et al.* [40] formulate the incentive mechanism between base station and clients as an auction game where the base station is an auctioneer and the clients are the sellers. As for the clients, it makes optimal communi-

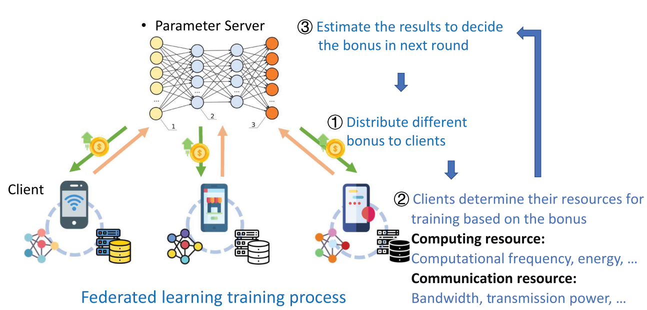


FIGURE 4. An illustration of incentive mechanism for federated learning driven by clients' resource allocation.

TABLE 3. Summary of incentive mechanism design driven by clients' resource allocation.

Approach	Literature	Applications	Orientation	Description
Stackelberg Game	[36]	Classification	Computational resource	Judiciously determine a subset of workers for each round's training to seek trade-off between heterogeneous workers' performance and the latency of completing the training
	[37]	IoT	Computational resource	Design incentive mechanism for enabling federated learning at network edge
	[38]	Classification	Communication resource	Constructing a communication-efficient cost model to build a novel crowdsourcing platform which considers the communication efficiency during model parameters exchange
Deep Reinforcement Learning (DRL)	[39]	Classification	Computational resource	Using DRL to find optimal trade-off between model training time and server's payment under dynamic network environment

cation resource allocation strategy so the energy cost is minimized while the delay constraint of federated learning is satisfied. For the base station, the client selection problem in the auction game is formulated as the social welfare maximization problem which is an NP-hard problem. Then a primal-dual greedy algorithm is proposed to solve the NP-hard problem. Meanwhile, the truthfulness and individual rationality of the proposed auction mechanism are also proved.

Similar to the above work, Pandey *et al.* propose a novel crowdsourcing platform by constructing a communication-efficient cost model that considers the communication efficiency during model parameters exchange [38]. The incentive mechanism is introduced as a value-based compensation strategy, such as a bonus, which is proportional to the level of federated learning participation. A two-stage Stackelberg game approach is adopted to solve the primal-dual optimization problem where each side maximizes its own benefit. Besides, they also present an admission control scheme for clients to ensure a certain level of local accuracy.

VI. FUTURE DIRECTIONS

Incentive mechanism design for federated learning is in its infancy. More works need to be done to guarantee sufficient number of clients for the rapidly proliferating machine learning applications. In this section, we discuss 3 possible future directions on incentive mechanism for federated learning, with respect to multi-party federated learning, incentive-driven federated learning, and secure federated learning.

A. INCENTIVE MECHANISM FOR MULTI-PARTY FEDERATED LEARNING

In current works, we can find that there is only one federated learning task publisher and multiple clients. The task publisher incentivizes the clients to participate in the federated learning through rewards. Different from this architecture, in multi-party machine learning, a group of parties cooperate on optimizing towards their own better models [41], [42]. To make multi-party machine learning practical, lots of works focus on preserving data privacy [43], [44] and learning performance in the learning process. However, incentive mechanism as one of the main critical issues has been ignored in previous works, which will significant reduce the effectiveness when putting this technique into practice [45]. Previous works usually let all the parties share the same global machine learning model without regarding each party's own contribution. Only when there are no conflicts among the parties can this mechanism works well. For example, the Gmail wants to use the users' experience data to increase their work efficiency. In this case, all the users are happy to contribute their data unreservedly since they can all benefit from such improvements, and there are no conflicts among the users.

When the parties are competing with one another, they maybe unwilling to participate in the federated learning since the other competitors can also benefit from their own contributions. For example, when several companies from the same field try to adopt the federated learning to level up their serving qualities. In this case, improving others' qualities can possible harm their own market share, especially for the large companies which have lots of high quality data. Such a cooperative and competitive relationships among the different parties pose an interesting challenge that prevents the multi-party learning approach from being applied to a wider range of environments. One of the possible solutions is to use incentive mechanism. As shown in Figure 5, client A is with lower contribution for federated learning, while client B is with higher contribution. In this case, in order to obtain higher serving quality, the client A wants to participate in the federated learning. However, in order to secure market share, client B does not want to be involved. With the incentive mechanism, the client A can participate in federated learning

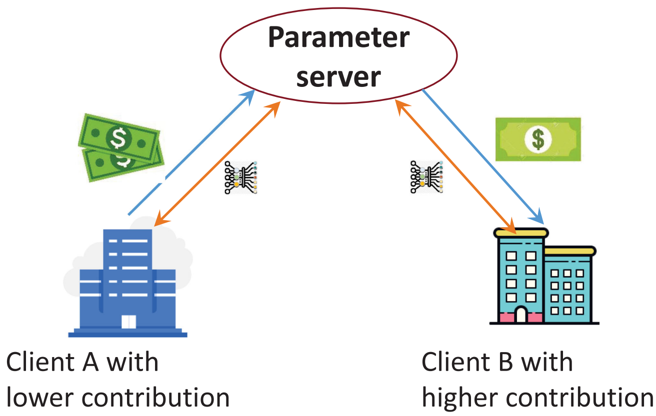


FIGURE 5. An illustration of incentive mechanism for multi-party federated learning.

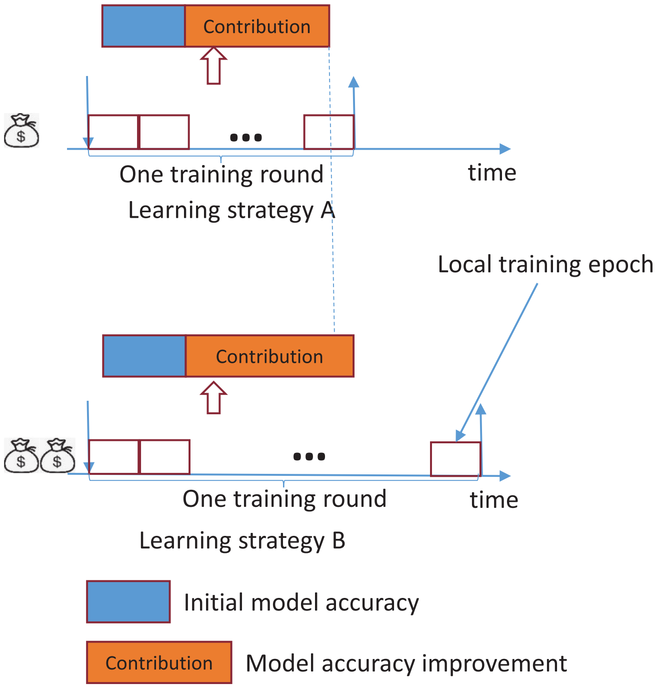
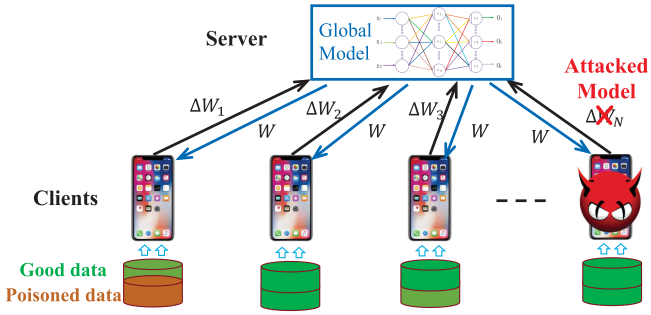
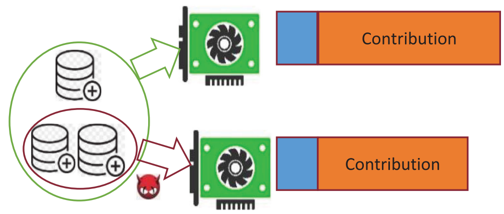


FIGURE 6. **An illustration of incentive-driven federated learning.**



(a) Attack.



(b) Aborting local training early.

FIGURE 7. **An illustration of security issues in federated learning.**

by paying rewards to the system. Then, the system can incentivize the client B to participate in federated learning by paying rewards to it. Recently, Chen [45] has designed the optimal truthful mechanism in the quasi-monotone utility setting, which provides a starting point for multi-party federated learning. We believe more research works will be conducted in this direction.

B. INCENTIVE-DRIVEN FEDERATED LEARNING

Current works are based on the training mode of federated learning, but they are not really combined with federated learning algorithm. For example, existing works assume that in each training round, the clients will train the local machine learning models with the same local epochs in each training round [25], [26], [39]. Hence, the incentive mechanism can only motivate the clients to participate in the federated learning, but cannot adjust the learning algorithm.

As shown in Figure 6, we can adjust the federated learning algorithm through carefully controlling the number of local epochs. With learning strategy A, the parameter server gives less money to the clients in each training round, then the clients train the local machine learning model with less local epochs and obtain less model accuracy improvement. With learning strategy B, the parameter server gives more money to the clients in each training round to drive them train the machine learning models with more local epochs, this makes the model accuracy promote higher. In learning strategy A, the clients train the local machine learning models with less local epochs, but with more training rounds. Learning strategy B, however, does the opposite. At the same time, some facts have shown that communication costs dominate the overhead of federated learning. Reducing the number of communication rounds as many as possible is an essential requirement for federated learning. Therefore, it is hard to judge whether learning strategy A is better than strategy B. How to carefully control the number of local epochs with the incentive-driven approaches is not only difficult but also challenging. We believe that with carefully design the incentive mechanism, the federated learning system could obtain a very good machine learning model. There is no work in this direction yet, and we believe that more researchers will be involved in the future.

C. SECURITY

Even though there are many works focusing on the incentive mechanism design for federated learning, yet they do not consider one of the critical issues, i.e., security. Considering a situation that clients may have malicious behaviors during federated learning. They may choose their inputs at will and thus generate incorrect gradients, aiming to mislead the federated learning process. As shown in Figure 7(a), there are two approaches to generate incorrect gradients: one is to inject the poisoned data into the training data, the other is to upload the poisoned model. With these two attacks, the federated learning will be biased. As shown in Figure 7(b), there is also another malicious behavior. In order to save their resource, the clients can train their local machine learning models with less training data, thus abort the local training early. By this, the clients will make less contribution for the federated learning system, and thus the approaches designed in [25], [26], [39] will not only waste more money, but also yield a lower-quality model.

The federated learning system is vulnerable to be attacked, which will seriously degrade the system performance. In order to keep a benign ecological environment for federated learning system, the incentive mechanism can be applied. With incentive mechanism, we can punish malicious clients, thus reduce their probability of doing evil. This is a new, but very important direction, and we believe that this is a very promising direction.

VII. CONCLUSION

Incentive mechanism is a key design element of novel federated learning systems. In this work, we extensively survey the state-of-the-art approaches and open up a few interesting future research directions. First, we begin with an introduction of how federated learning and incentive mechanism work in general. Then, we provide detailed reviews, analysis, and comparisons of solutions for emerging implementation challenges in the incentive mechanism design for federated learning. The issues include model contribution, clients' reputation, and resource allocation. Finally, we propose a few interesting future research directions include incentive mechanism design for multi-party federated learning, incentive-driven federated learning, and security guarantee with incentive mechanism.

In conclusion, incentives play an essential role in federated learning systems, as they feed the system on sufficient number of clients such that the federated learning systems can actually work. With the rapidly proliferating machine learning applications, the development of effective and efficient incentive mechanism for federated learning is a new and vibrant filed. From this survey, we expect more and more researchers to devote themselves to this field.

ACKNOWLEDGMENTS

This work was supported in part by the funding from Hong Kong RGC Research Impact Fund (RIF) under the Project No. R5060-19 and R5034-18, in part by General Research Fund (GRF) under the Project No. 152221/19E and 15220320/20E, in part by Collaborative Research Fund (CRF) under the Project No. C5026-18G, in part by the National Natural Science Foundation of China under Grant 61872310, and in part by JSPS Grants-in-Aid for Scientific Research under Grant JP19K20258.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, early access, Mar. 4, 2019, doi: 10.1109/TETC.2019.2902661.

[3] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.

[4] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: A distributed learning framework and applications," *IEEE Wireless Commun.*, 2020.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[6] H. B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist. Conf.*, 2017, pp. 1273–1282.

[7] S. Wang *et al.*, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 63–71.

[8] Z. M. Fadlullah and N. Kato, "HCP: Heterogeneous computing platform for federated learning based collaborative content caching towards 6G networks," *IEEE Trans. Emerg. Topics Comput.*, early access, Apr. 8, 2020, doi: 10.1109/TETC.2020.2986238.

[9] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 38–67, First Quarter 2020.

[10] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, 2017.

[11] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open J. Comput. Society*, vol. 1, pp. 45–61, May 6, 2020.

[12] K. Powell, "Nvidia clara federated learning to deliver ai to hospitals while protecting patient data," Accessed: Dec. 1, 2019. [Online]. Available: https://blogs.nvidia.com/blog/2019/12/01/clara-federated-learning/

[13] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020.

[14] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 587–601.

[15] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 173–184.

[16] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A stackelberg game approach," *Comput. Netw.*, vol. 129, pp. 399–409, 2017.

[17] P. Li and S. Guo, "Incentive mechanisms for device-to-device communications," *IEEE Netw.*, vol. 29, no. 4, pp. 75–79, Jul./Aug. 2015.

[18] Y. Zhan, Y. Xia, J. Zhang, and Y. Wang, "Incentive mechanism design in mobile opportunistic data collection with time sensitivity," *IEEE Internet of Things J.*, vol. 5, no. 1, pp. 246–256, Feb. 2018.

[19] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2577–2586.

[20] H. Yu *et al.*, "A Sustainable incentive scheme for federated learning," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 58–69, Jul./Aug. 2020.

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

[22] R. Zeng, S. Zhang, J. Wang, and X. Chu, "FMore: An incentive scheme of multi-dimensional auction for federated learning in MEC," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 278–288.

[23] Y. Zhan, P. Li, K. Wang, S. Guo, and Y. Xia, "Big data analytics by CrowdLearning: Architecture and mechanism design," *IEEE Netw.*, vol. 34, no. 3, pp. 143–147, May/Jun. 2020.

[24] Y. Zhan, J. Zhang, P. Li, and Y. Xia, "Crowdtraining: Architecture and incentive mechanism for deep learning training in the internet of things," *IEEE Netw.*, vol. 33, no. 5, pp. 89–95, Sep./Oct. 2019.

[25] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[26] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, early access, Nov. 08, 2019, doi: 10.1109/TDSC.2019.2952332.

[27] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: A blockchain for auditable federated learning with trust and incentive," in *Proc. 5th Int. Conf. Big Data Comput. Commun.*, 2019, pp. 151–159.

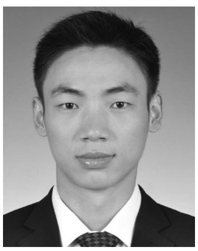
[28] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, Jan. 2020.

[29] N. Ding, Z. Fang, and J. Huang, "Incentive mechanism design for federated learning with multi-dimensional private information," in *Proc. 18th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw.*, 2020, pp. 1–8.

[30] R. Jia *et al.*, "Towards efficient data valuation based on the shapley value," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1167–1176.

[31] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, "Collaborative machine learning with incentive-aware model rewards," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8927–8936.

[32] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, “Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system,” 2019, *arXiv: 1906.10893*.
[33] M. H. ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, “Towards blockchain-based reputation-aware federated learning,” in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2020, pp. 183–188.
[34] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, “Reliable federated learning for mobile networks,” *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.
[35] M. Chen, Y. Liu, W. Shen, Y. Shen, P. Tang, and Q. Yang, “Mechanism design for multi-party machine learning,” 2020, *arXiv: 2001.08996*.
[36] 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.
[37] L. Khan, S. Pandey, N. H Tran, W. Saad, M. N. Nguyen, and C. S. Hong, “Federated learning for edge networks: Resource optimization and incentive mechanism,” *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, 2020.
[38] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, “A crowdsourcing framework for on-device federated learning,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, May 2020.
[39] Y. Zhan and J. Zhang, “An incentive mechanism design for efficient edge learning by deep reinforcement learning,” in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2489–2498.
[40] T. H. T. Le *et al.*, “An incentive mechanism for federated learning in wireless cellular network: An auction approach,” 2020, *arXiv: 2009.10269*.
[41] Y. Hu, D. Niu, J. Yang, and S. Zhou, “FDML: A collaborative machine learning framework for distributed features,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2232–2240.
[42] V. Smith, C. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.
[43] M. Abadi *et al.*, “Deep learning with differential privacy,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.
[44] R. Yonetani, V. N. Boddeti, K. M. Kitani, and Y. Sato, “Privacy-preserving visual learning using doubly permuted homomorphic encryption,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2059–2069.
[45] M. Chen, Y. Liu, W. Shen, Y. Shen, P. Tang, and Q. Yang, “Mechanism design for multi-party machine learning,” 2020, *arXiv:2001.08996*.



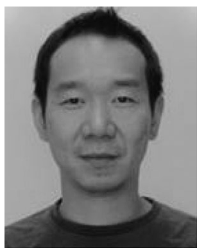
YUFENG ZHAN is an assistant professor with the School of Automation, Beijing Institute of Technology. Prior to join BIT, he was a post-doctoral fellow with the Department of Computing, Hong Kong Polytechnic University. His research interests include networking systems, game theory, and machine learning.

JIE ZHANG is currently working toward the PhD degree with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. Her current research interests include edge computing, federated learning, and deep reinforcement learning.

ZICONG HONG is currently working toward the PhD degree with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His current research interests include Blockchain, game theory, Internet of Things, and edge/cloud computing.

LEIJIE WU is currently working toward the PhD degree with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His current research interests include distributed machine learning, game theory, and deep reinforcement learning.

PENG LI (Senior Member, IEEE) received the BS degree from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the MS and PhD degrees from the University of Aizu, Aizuwakamatsu, Japan, in 2009 and 2012, respectively. He is currently an associate professor with the University of Aizu, Japan. His research interests mainly focus on cloud/edge computing, Internet-of-Things, machine learning systems, as well as related wired and wireless networking problems. He has published more than 100 technical papers on prestigious journals and conferences. He won the Young Author Award of IEEE Computer Society Japan Chapter, in 2014. He won the Best Paper Award of IEEE TrustCom 2016. He supervised students to win the First Prize of IEEE ComSoc Student Competition, in 2016. He is the editor of the *IEICE Transactions on Communications* and the *IEEE Open Journal of the Computer Society*.



SONG GUO (Fellow, IEEE) is a full professor with the Department of Computing, Hong Kong Polytechnic University. He also holds a Changjiang Chair Professorship awarded by the Ministry of Education of China. His research interests are mainly in the areas of big data, edge AI, mobile computing, and distributed systems. He has published more than 500 papers in major journals and conferences and been recognized as a Highly Cited Researcher (Web of Science). He is the recipient of more than 12 best paper awards from IEEE/ACM conferences, journals and technical committees. He is the editor-in-chief of the *IEEE Open Journal of the Computer Society* and the chair of IEEE Communications Society (ComSoc) Space and Satellite Communications Technical Committee. He has served on IEEE ComSoc Board of Governors, IEEE Computer Society on Fellow Evaluation Committee, and editorial board of a number of prestigious international journals like the *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Internet of Things Journal*, etc. He has also served as chair of organizing and technical committees of many international conferences. He is an ACM distinguished member.