Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory

Jiawen Kang, Zehui Xiong, *Student Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, Shengli Xie, *Fellow, IEEE*, and Junshan Zhang, *Fellow, IEEE*

***Abstract*—Federated learning is an emerging machine learning technique that enables distributed model training using local datasets from large-scale nodes, e.g., mobile devices, but shares only model updates without uploading the raw training data. This technique provides a promising privacy preservation for mobile devices while simultaneously ensuring high learning performance. The majority of existing work has focused on designing advanced learning algorithms with an aim to achieve better learning performance. However, the challenges, such as incentive mechanisms for participating in training and worker (i.e., mobile devices) selection schemes for reliable federated learning, have not been explored yet. These challenges have hindered the widespread adoption of federated learning. To address the above challenges, in this article, we first introduce reputation as the metric to measure the reliability and trustworthiness of the mobile devices. We then design a reputation-based worker selection scheme for reliable federated learning by using a multiweight subjective logic model. We also leverage the blockchain to achieve secure reputation management for workers with nonrepudiation and tamper-resistance properties in a decentralized manner. Moreover, we propose an effective incentive mechanism combining reputation with contract theory to motivate high-reputation mobile devices with high-quality data to participate in model learning. Numerical results clearly indicate that the proposed schemes are efficient for reliable federated learning in terms of significantly improving the learning accuracy.**

***Index Terms*—Blockchain, contract theory, federated learning, mobile networks, reputation, security and privacy.**

Manuscript received June 12, 2019; revised August 15, 2019; accepted September 2, 2019. Date of publication September 11, 2019; date of current version December 11, 2019. This work was supported in part by WASP/NTU under Grant M4082187 (4080), in part by Singapore Ministry of Education (MOE) Tier 1 under Grant 2017-T1-002-007 RG122/17, in part by MOE Tier 2 under Grant MOE2014-T2-2-015 ARC4/15, in part Singapore-Israel NRF-ISF under Grant NRF2015-NRF-ISF001-2277, in part by EMA Energy Resilience under Grant NRF2017EWT-EP003-041, and in part by the Programs of NSFC under Grant 61973087 and Grant 61703113. The work was presented in part at the 16th IEEE Asia Pacific Wireless Communications Symposium 2019. *(Corresponding author: Zehui Xiong.)*

J. Kang, Z. Xiong, and D. Niyato are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: kavinkang@ntu.edu.sg; zxiong002@e.ntu.edu.sg; dniyato@ntu.edu.sg).

S. Xie is with the Guangdong Key Laboratory of IoT Information Technology, Ministry of Education, Guangdong University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Intelligent Detection and Internet of Manufacturing Things, Ministry of Education, Guangdong University of Technology, Guangzhou 510006, China (e-mail: shlxie@gdut.edu.cn).

J. Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: junshan.zhang@asu.edu).

Digital Object Identifier 10.1109/JIOT.2019.2940820

I. Introduction

With the rapid development of machine learning technologies, many novel mobile applications, e.g., autonomous driving, are emerging with machine learning which brings great service experience to mobile users [1]. Although the machine learning techniques significantly improve the performance of mobile applications, most of the machine learning techniques need to aggregate massive user data with personal information into a central server to perform model training. This causes excessive computation and storage cost, and the mobile devices also suffer from serious privacy leakage risk [2]. To address these challenges, federated learning as an emerging distributed machine learning paradigm has been introduced to allow the mobile devices to collaboratively train a global model in a decentralized manner. The mobile devices only need to iteratively send local model updates trained on their local raw data to the task publisher instead of uploading the raw data outside, thus decoupling the machine learning from acquiring, storing and training data in a central server [3].

Despite the aforementioned great benefits, federated learning is still facing critical challenges. On the one hand, existing studies have an optimistic assumption that all the mobile devices contribute their resources unconditionally [4], [5], which is not practical in the real world due to the resource costs incurred by model training [6]. Without well-designed economic compensation, the self-interested mobile devices are reluctant to participate in model training [3], [5], [7]. Therefore, it is necessary to design an effective incentive mechanism for stimulating mobile devices to become workers for federated learning tasks [1], [8]. On the other hand, an unreliable worker may perform intentionally or unintentionally undesirable behaviors to mislead a global model training of a federated learning task. For intentional behaviors, the worker may launch a poisoning attack that sends malicious updates to influence the global model parameters leading to the failure of current collaborative learning mechanism. Moreover, much more dynamic mobile networking environments indirectly cause some unintentional behaviors of mobile devices. Due to the high mobility or energy constraints, the worker may nondeliberately update some low-quality parameters that adversely affect federated learning tasks. Thus, it is of vital importance to develop efficient schemes to select workers without unreliable local model updates for the federated

learning. Furthermore, these two challenges cannot be solved separately due to their couplings [3].

Accordingly, we are motivated to address incentive issues about high-quality worker selection and reliable model training. To achieve high-quality worker selection, a fair metric is essential to evaluate the reliability of a worker. Previous work has utilized reputation as the metric to rate the reliability or trustworthiness of an entity in certain activities according to its past behaviors [9]–[12]. Inspired by this, we also apply reputation as the metric to assess the reliability of a federated learning worker candidate, thus ensuring reliable worker selection. Note that the high-reputation workers bring high-quality data (i.e., high-accuracy and reliable data) to model training and generate reliable local model updates for federated learning tasks [11], [13]. Therefore, for better performance of the federated learning tasks, every task publisher chooses high-reputation worker candidates that have high-accuracy and reliable local data as the workers to reduce the impacts from unreliable attackers [4]. Each task publisher calculates reputation opinions on interacting workers by using a subjective logic model, in which direct reputation opinions generated from past interactions and indirect reputation opinions from other task publishers are integrated into a compositive reputation for worker selection.

To ensure secure reputation management and reliable reputation calculation, unlike conventional centralized management structure with the risk of single point of failure, we manage reputation opinions from task publishers in a decentralized manner through a consortium blockchain with the properties of nonrepudiation and tamper-resistance. The consortium blockchain is a more efficient, practical blockchain technology with a light-weight and faster consensus process controlled by preselected miners. With the above advantages, the consortium blockchain is especially suitable for reputation management in a mobile network with federated learning [14].

Moreover, to stimulate the workers to join model training and share their resources, the task publisher employs contract theory to design an incentive mechanism that specifies the contributed resources (e.g., data, computation, and communication resources) and the corresponding rewards. The higher-reputation workers that have larger accuracy of local data or more reliable behaviors can obtain more rewards from the task publisher. Each worker chooses its desired contract item to maximize its profit [6].

The main contributions of this article are listed as follows.

1) We introduce reputation as a fair metric to select reliable workers for federated learning to defend against unreliable model updates in mobile networks.
2) We utilize a multiweight subjective logic model to efficiently calculate the worker reputation and manage the reputation in a decentralized manner through the consortium blockchain.
3) We design an effective incentive mechanism using contract theory to stimulate high-reputation workers with high-quality data to participate in model training in order to prevent poisoning attacks in federated learning.
4) We perform experiments on a real-world dataset to show that the proposed model is practically applicable to real federated learning applications. The numerical results validate that the proposed incentive mechanism outperforms the traditional ones. The blockchain scheme can ensure accurate reputation calculation, thus significantly improving the learning accuracy.

The rest of this article is organized as follows. We present the related work in Section II and the preliminary in Section III. Section IV describes the reputation-based worker selection scheme based on consortium blockchain. Section V illustrates the reputation calculation using the multiweight subjective logic model. An incentive mechanism for reliable worker selection using contract theory is proposed in Section VI, followed by the performance evaluation in Section VII. Finally, we summarize this article in Section VIII.

II. Related Work

Google first introduced the federated learning technology and also used it to design a virtual keyboard application for smart phones named Gboard [15]. Along with this direction, the concept, architecture and potential applications about federated learning are further discussed in [1]. From the perspective of optimization, Wang *et al.* [16] combined deep reinforcement learning and the federated learning framework into mobile edge systems to upgrade the federated learning architecture to achieve mobile edge computing, caching, and communication optimization. Tran *et al.* [3] formulated an optimization problem of federated learning over wireless networks to obtain optimal learning time, accuracy level, and energy cost. Samarakoon *et al.* [17] employed federated learning to estimate the tail distribution of network-wide queues, thus minimizing network wide power while ensuring ultrareliable and low-latency vehicular communication. Anh *et al.* [8] utilized a deep Q learning algorithm to solve the optimal data and energy management problems of federated learning without prior knowledge of network dynamics. Considering clients with heterogeneous resources, Nishio and Yonetani [18] proposed a client selection scheme for federated learning based on a greedy algorithm.

From the perspective of security, Shayan *et al.* [4] stated that mobile devices may perform poisoning attacks on model updates or information leakage attacks against a target mobile device. They utilized the reject on negative influence (RONI) defense to remove the poisoning model updates and used a differential privacy scheme for privacy preservation. The model updates were stored and aggregated in a blockchain platform. Similarly, Kim *et al.* [5] stored and verified local model updates using blockchain, but further analyzed the end-to-end learning completion latency to optimize the block generation rate. Fung *et al.* [19] introduced the Sybil-based poisoning attacks and then identified these attacks according to client contribution similarity.

The above work specifically focused on optimizing the performance of learning algorithms, e.g., learning time or training security, in federated learning. However, the majority of the work made the same assumption that all the mobile devices contribute data, communication or computation resources unconditionally for the federated learning [1].

TABLE I
MAIN SYMBOLS

Notation	Definition
Φ	A shared global model
s_n	Data size of data samples
$\ell_n(.)$	The local loss function of a model device n
t_y	The y-th time slot in a time window
$\{b_{i\to j}, d_{i\to j}, u_{i\to j}\}$	Reputation opinion vector $\omega_{i\to j}$ of task publisher i for worker j
$T_{i\to j}$	Reputation of task publisher i for worker j
$\alpha_i^{t_y}$, $\beta_i^{t_y}$	The number of positive / negative interactions in t_y between task publisher i and worker j
κ, η	Weight of positive / negative interactions
$\vartheta(t_y)$	Freshness fading function of interaction events in a time slot t_y
$\theta_n, n \in \{1, \ldots, N\}$	The types of workers
f_n	CPU-cycle frequency of type-n worker
R_n	Reward for type-n worker
c_n	The CPU cycles for training one data sample
E_n^{cmp} / T_n^{cmp}	CPU energy consumption / computation time of a local model iteration for worker n
E_n^{com} / T_n^{com}	Energy consumption / Transmission time of local model update transmission for worker n
l	Unit resource cost of rewards for the workers
μ	Weight parameter about energy consumption
p_n	Probability of a worker belonging to type-n

Although Nishio and Yonetani [18] considered the resource limitation and worker selection issues, the reliability of workers was ignored. Resource-rich workers are reluctant to join model training without proper incentive mechanisms. Furthermore, the incentive mechanisms and worker selection issues are coupled and should be solved jointly for federated learning. In this article, we therefore consider both incentive mechanism and selection scheme for reliable workers. To the best of our knowledge, this is the first work to employ contract theory in model training for federated learning, which brings desirable economic and resource allocation benefits.

III. PRELIMINARIES

In this section, we briefly introduce the basics of federated learning followed by the system model and the adversary model under our consideration. The main mathematical notations used in this article are listed in Table I.

A. Federated Learning

Federated learning is a promising distributed privacy-preserving machine learning technique that enables mobile devices to collaboratively train a shared global model without the need of uploading private local data to a central server. Each mobile device obtains a shared global model Φ from a task publisher and trains the model over its local data. Then, the mobile device uploads the new weights or gradients (i.e., local model update) to the task publisher for updating the global model. Specifically, each mobile device n has a local training dataset with s_n data samples for federated learning. The total size of data samples from N mobile devices is $\sum_{n=1}^{N} s_n = S$. The federated learning aims to optimize a global loss function $\ell(\Phi)$ through minimizing the weighted average of every mobile device n's local loss function $\ell_n(\Phi)$ [20] on its local dataset, i.e.,

$$\min_{\Phi} \ell(\Phi) = \sum_{n=1}^{N} \frac{s_n}{S} \ell_n(\Phi), \text{ where } \ell_n(\Phi) = \frac{1}{s_n} \sum_{i \in s_n} f_i(\Phi). \quad (1)$$

Here, $f_i(\Phi)$ is the loss function of sample data i in the local dataset for mobile device n [3].

At a global training iteration t, every mobile device computes its average gradient Λ_n on its local dataset with the current global model $\Phi^{(t)}$ by using optimization algorithms. Without loss of generality, we utilize the stochastic gradient descent (SGD) algorithm which iteratively selects a batch of training samples to calculate their gradients against $\Phi^{(t)}$ and takes gradient steps in the direction that minimizes $\ell_n(\Phi)$ [4], [15]. Given a learning rate $\hbar_n$ of the mobile device n, the local model update is expressed by

$$\Phi_n^{(t+1)} = \Phi^{(t)} - \hbar_n \Lambda_n. \quad (2)$$

Thus, the task publisher updates the shared global model $\Phi^{(t+1)}$ through a weighted aggregation of all the local model updates [15], which is denoted as

$$\Phi^{(t+1)} = \sum_{n=1}^{N} \frac{s_n}{S} \Phi_n^{(t+1)}. \quad (3)$$

Note that, according to (1), the high-quality mobile devices with high-accuracy and reliable local training data can lead to faster convergence of the local loss function $\ell_n(\Phi)$ and the global loss function $\ell(\Phi)$ [3]. Both the local model update Φ_n of the worker in (2) and the shared global model update Φ in (3) can be quicker to converge to the target value with fewer iterations. Consequently, the training time and energy consumption of a worker in a global iteration decrease. Therefore, high-quality mobile devices with high accuracy and reliable local training data can significantly improve the learning efficiency of federated learning, e.g., less training time and less energy consumption [3], [4], [18].

B. System Model

Fig. 1 shows a federated learning system with a consortium blockchain, including an application layer and a blockchain layer. For the application layer, we consider a universal mobile network consisting of wireless communication infrastructures (e.g., roadside units in vehicular networks or base stations with edge computing in cellular networks) and a set of mobile devices (such as vehicles or mobile phones). The widely deployed communication infrastructures can be treated as edge nodes. The mobile devices equipped with advanced computation and communication devices can not only generate diverse user data from mobile applications but also collect a lot of sensing data. In mobile networks, multiple task publishers with federated learning tasks execute model training based on these data from the mobile devices, without data collection, for the sake of privacy protection. Each task publisher designs contract items for incentivizing reliable mobile devices acting as workers for model training (step *a* in Fig. 1). Every worker iteratively trains a shared global model Φ by its local data and generates local model updates Φ_n (steps *b, c, d* in Fig. 1).

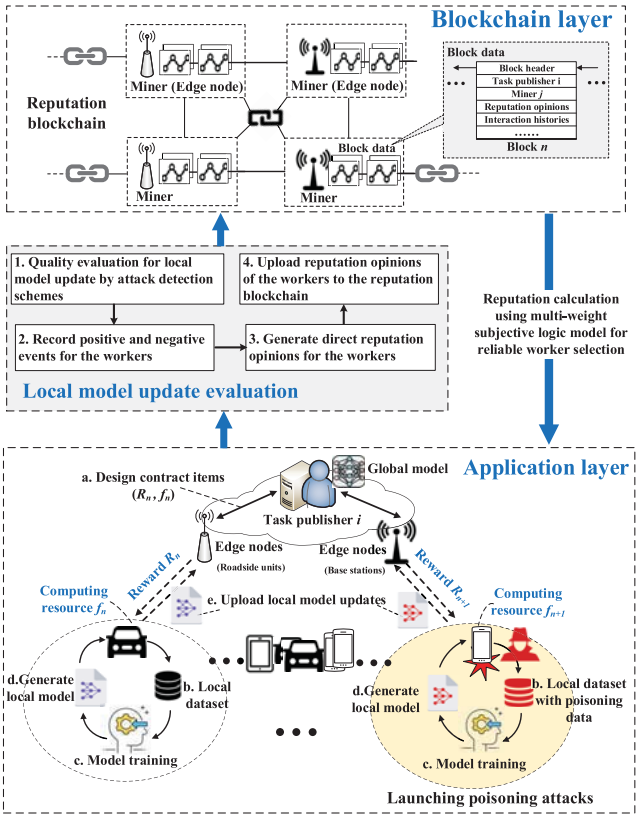


Fig. 1. System model of reliable federated learning based on a consortium blockchain.

Then, all the workers upload their local model updates to the task publisher for updating the global model (step *e* in Fig. 1). The training process is repeated until the accuracy of the global model achieves a predefined, desirable value. The widely distributed edge nodes enable the workers to communicate with the task publishers in a timely fashion. The task publishers evaluate the quality of the local model updates and generate reputation opinions for the interacted workers according to their model updates and training behaviors. More details about the local model update evaluation and reputation calculation are given in Sections IV and V, respectively. These reputation opinions are maintained by a consortium blockchain named *reputation blockchain* with the properties of tamper-proof and nonrepudiation, and are shared among the task publishers.

In the blockchain layer, the edge nodes are easy to be connected by the mobile devices and the federated learning task publishers, thus working as the preselected miners of consortium blockchains because of having sufficient computation and storage resources [9], [11], [21]. The details of the preselected miner selection scheme for the reputation blockchain are not the focus of this article. Nonetheless, we can straightforwardly adopt some of the most popular ones: random selection [5] and reputation-based miner selection [9]. The reputation opinions are stored into data blocks after verification by the miners via performing a consensus algorithm. Due to the decentralized and tamper-proof nature of blockchains, the reputation opinions in the data blocks are persistent and transparent evidence even when a dispute and destruction occurs [9]. For a specific mobile device, the task publisher integrates its direct reputation opinion with the latest indirect reputation opinions from other task publishers to generate a compositive reputation value for the mobile devices. The reputation value is an important metric for reliable worker selection during federated learning.

C. Adversary Model

For federated learning, due to the openness and complexity of mobile network architectures, mobile devices acting as workers may perform malicious and unreliable model updates. On the one hand, the unreliable model updates may be generated from the following intentional reasons. Malicious or tampered devices train the data with deceptive information or transmit the data through unsafe communication channels [2]. Hence, a malicious data owner (i.e., malicious worker) may intentionally launch serious attacks, e.g., poisoning attacks [4]. The poisoning attack means that the malicious workers deliberately inject poisonous data points into training datasets or modify the training datasets to degrade the accuracy of training data, thus increasing the probability of incorrect classification and manipulating the results of their local model updates in (2). On the other hand, the workers may unintentionally generate unreliable local updates caused by unreliable wireless communication channel conditions or poor data sensing environments due to high mobility or energy constraints [18]. As a result, both the intentional and unintentional behaviors degrade the local data quality and local model update quality, hence negatively affecting the accuracy and convergence time of the global model in (3). Therefore, it is essential to design a reliable worker selection scheme for federated learning.

IV. Reputation-Based Worker Selection Scheme With Consortium Blockchain

To address the aforementioned challenges, as shown in Fig. 1, we design a reputation-based worker selection scheme for reliable federated learning and also utilize a consortium blockchain to manage the reputation. The reputation management mainly includes reputation calculation by using multiweight subjective logic model and secure reputation storage in a decentralized manner through consortium blockchain technologies. More details of the worker selection scheme are given as follows.

1) *Step 1 (Publish Federated Learning Tasks and Contract Items):* Task publishers first design contract items according to data and computation resource requirements of their federated learning tasks. Each task publisher broadcasts its federated learning task with specific resource requirements (e.g., data types, data sizes, and accuracy, time range, and CPU cycles) and the contract items to mobile devices. The mobile devices satisfying the requirements may become the model training worker candidates to join the federated learning tasks, and also give a response including resource information back to the task publisher.
2) *Step 2 (Calculate Candidate Reputation):* The task publisher chooses the qualified worker candidates according to the resource information. Then, the task publisher

calculates reputation values of the worker candidates by a multiweight subjective logic model according to: a) direct reputation opinions from interaction histories and b) indirect reputation opinions from other task publishers (i.e., recommended reputation opinions). The recommended reputation opinions are stored and managed on the open-access reputation blockchain. The reputation blockchain is a public ledger that records reputation opinions of worker candidates into data blocks. For each worker candidate, the task publisher first downloads the latest recommended reputation opinions from the reputation blockchain. Thus, the task publisher combines its direct reputation opinion with the recommended reputation opinions to generate a compositive value as the final reputation for each candidate. More details about the reputation calculation are presented in Section V.

3) *Step 3 (Select Workers for Federated Learning):* After reputation calculation, the worker candidates with reputation larger than a threshold can be selected as the workers. These workers make their own optimal decisions to select a contract item given by the task publisher according to their types related to quality of local dataset and resource conditions. The quality of local dataset directly determines the quality of local model updates [4]. The details about contract designing are given in Section VI.
4) *Step 4 (Perform Federated Learning and Evaluate Quality of Local Model Updates):* After worker selection, the federated learning tasks can be trained by different optimization algorithms, e.g., SGD. Specifically, an initial SGD model (i.e., initial parameters) is randomly chosen from predefined ranges as the shared global model. After receiving this model, the workers collaboratively train the model over their own local data and upload their local model updates to the task publisher. To evaluate the reliability of the local model updates, the task publisher performs quality evaluation of local model updates through attack detection schemes, e.g., reject on negative influence (RONI) scheme in [4] for independent and identically distributed (IID) data scenarios and FoolsGold scheme in [19] for non-IID data scenarios, to identify the poisoning attacks and unreliable workers (step 1 in Fig. 1). Here, RONI is a typical poisoning attack detection scheme that validates a local model update by comparing the effects with and without the local model update on a predefined database from the task publisher. If the performance of the local model update on the database degrades beyond a specified threshold given by the system, this local model update will be rejected when integrating all local model updates [4]. The FoolsGold scheme identifies unreliable workers according to the gradient update diversity of local model updates in non-IID federated learning in that the training data of each worker has a unique distribution. The unreliable workers can be detected as they repeatedly upload similar-looking gradients as local model updates in each iteration [19]. With the above unreliable worker and attacker detection schemes, the task publisher can remove unreliable local model updates from the unreliable workers as well as malicious updates from the poisoning attacks. The task publisher integrates all the reliable local model updates into an average value and sets the average value as the new global model for the next iteration. The task publisher pushes this new model to the selected workers for the next model iteration until the latest global model satisfies a predefined convergence condition. Then the workers obtain rewards from the task publisher according to the preset rewards in the contact items based on resource contribution and model training behaviors [4], [5]. In every iteration, the interaction either with unreliable workers or with poisoning attackers is treated as a negative interaction and recorded by the task publisher. Finally, the task publisher generates the direct reputation opinions for all the workers in the federated learning task according to past interactions (steps 2 and 3 in Fig. 1).
5) *Step 5 (Update Reputation Opinions to the Reputation Blockchain):* After finishing a federated learning task, the task publisher updates its direct reputation opinions for the interacted workers according to the interaction histories (step 4 in Fig. 1). These reputation opinions with digital signatures for the workers are recorded as "transactions" and uploaded to the miners of the reputation blockchain. The miners put the reputation opinions into a data block and add the block to the reputation blockchain after block verification and executing consensus schemes, e.g., practical Byzantine fault tolerance (PBFT). Finally, all the task publishers can choose reliable workers with high reputations for their federated learning tasks with the help of the reputation blockchain.

V. Reputation Calculation Using Multiweight Subjective Logic

Since high-reputation workers with high-accuracy and reliable training data play vital roles in model training process, efficient and accurate reputation calculation is essential for reliable federated learning. In this section, we apply a subjective logic model to generate compositive reputation values of worker candidates. The subjective logic is a widely adopted framework of probabilistic reasoning that evaluates trustworthiness or reliability level of different entities. The subjective logic uses the term "opinion" to denote the representation of a subjective belief through positive, negative, and uncertainty statements [10], and can also combine and relate different opinions from a large amount of logical operators. In this article, to obtain more accurate reputation values of worker candidates, every task publisher combines its direct reputation opinions with the indirect reputation opinions to generate the compositive reputation values for the candidates.

A. Reputation Opinion Representation for Subjective Logic

For a time window with a series of time slots $\{t_1, \ldots, t_y, \ldots, t_Y\}$, the reputation opinions of task publisher i for worker j in a time slot t_y, e.g., 60 min, can be expressed by a tuple vector $\Upsilon_{i\to j}^{t_y} := \{b_{i\to j}^{t_y}, d_{i\to j}^{t_y}, u_{i\to j}^{t_y}\}$. Here, $b_{i\to j}^{t_y}$, $d_{i\to j}^{t_y}$,

and $u_{i\to j}^{t_y}$ are belief, disbelief and uncertainty, respectively [22]. $b_{i\to j}^{t_y}+d_{i\to j}^{t_y}+u_{i\to j}^{t_y}=1$, and $b_{i\to j}^{t_y}, d_{i\to j}^{t_y}, u_{i\to j}^{t_y}\in[0,1]$. Based on the subjective logic model [9], [12], we can obtain

$$\begin{cases} b_{i\to j}^{t_y}=\left(1-u_{i\to j}^{t_y}\right)\frac{\alpha_i^{t_y}}{\alpha_i^{t_y}+\beta_i^{t_y}} \\ d_{i\to j}^{t_y}=\left(1-u_{i\to j}^{t_y}\right)\frac{\beta_i^{t_y}}{\alpha_i^{t_y}+\beta_i^{t_y}} \\ u_{i\to j}^{t_y}=1-q_{i\to j}^{t_y}. \end{cases} \tag{4}$$

${\alpha_i}^{t_y}$ (${\beta_i}^{t_y}$) is the number of positive (negative) interactions during the time slot t_y. The task publisher treats a training iteration as a positive interaction event between itself and a worker if the publisher perceives that the local model update provided by the worker is useful, trusted and reliable by verification using attack detection schemes presented in step 4 of Section IV, and vice versa. $q_{i\to j}^{t_y}$ represents the success probability of data packet transmission, i.e., the communication quality, that affects the uncertainty of the reputation opinion. According to the reputation opinion vector, the direct reputation value of the task publisher i for the worker j in a time slot t_y is denoted as

$$T_{i\to j}^{t_y}=b_{i\to j}^{t_y}+au_{i\to j}^{t_y} \tag{5}$$

where $a\in[0,1]$ is the coefficient representing the degree of uncertainty effect on reputation [9].

Note that the reputation opinions are affected by many factors. Traditional subjective logic (TSL) is evolved toward multiweight subjective logic when taking different factors into consideration. Here, we consider the following factors to calculate the reputation opinions.

1) *Interaction Effects:* There exist positive and negative interaction results of the interaction events through quality evaluation of local model updates presented in step 4 of Section IV. The positive interactions increase the reputation of worker candidates, and vice versa. To discourage the negative interaction events, the negative interactions have a higher weight on the reputation calculation than the positive interactions. We denote the weights of positive and negative interactions as κ and η, respectively. $\kappa\le\eta$ and $\eta+\kappa=1$. Hence, the expression in (4) can be rewritten as

$$\begin{cases} b_{i\to j}^{t_y}=q_{i\to j}^{t_y}\frac{\kappa\alpha_i^{t_y}}{\kappa\alpha_i^{t_y}+\eta\beta_i^{t_y}} \\ d_{i\to j}^{t_y}=q_{i\to j}^{t_y}\frac{\eta\beta_i^{t_y}}{\kappa\alpha_i^{t_y}+\eta\beta_i^{t_y}} \\ u_{i\to j}^{t_y}=1-q_{i\to j}^{t_y}. \end{cases} \tag{6}$$

2) *Interaction Freshness:* The trustworthiness of a worker changes with time and a worker will not be always trusted and reliable during interactions between task publishers and workers. Recent interaction events with more freshness have larger weight than past events. To reflect the time effect on reputation, a freshness fading function is defined to illustrate the freshness of interaction events: $\vartheta(t_y)=\vartheta_y=z^{Y-y}$, where $z\in(0,1)$ is a given fade parameter about event freshness and $y\in[1,Y]$ is the time slot y that determines the fade degree of the event freshness. Therefore, the reputation opinion and the reputation value of the task publisher i for the worker j in a time window are denoted as

$$\begin{cases} b_{i\to j}=\frac{\sum_{y=1}^{Y}\vartheta_y b_{i\to j}^{t_y}}{\sum_{y=1}^{Y}\vartheta_y} \\ d_{i\to j}=\frac{\sum_{y=1}^{Y}\vartheta_y d_{i\to j}^{t_y}}{\sum_{y=1}^{Y}\vartheta_y} \\ u_{i\to j}=\frac{\sum_{y=1}^{Y}\vartheta_y u_{i\to j}^{t_y}}{\sum_{y=1}^{Y}\vartheta_y} \end{cases} \tag{7}$$

and $T_{i\to j}=([\sum_{y=1}^{Y}\vartheta_y T_{i\to j}^{t_y}]/[\sum_{y=1}^{Y}\vartheta_y])$.

B. Weighting Reputation Opinions From Recommenders

Each task publisher has its own interacted workers in mobile networks. Similar to social networks, the more commonly interacted workers between two task publishers bring larger credibility of indirect reputation opinions for them. The reputation opinions of every task publisher for the workers are denoted as an individual vector. The similarity of reputation opinions between the task publishers can be measured by the similarity of the vectors through using the amendatory cosine function. To measure the credibility of indirect reputation opinions, we define a *similarity factor* between the task publisher i and the task publisher x as weight by using an amendatory cosine function as follows [23]:

$$\mathrm{Sim}(i,x)=\frac{\sum_{j\in\mathbb{C}}\left(D_{i\to j}-\overline{D}_i\right)\left(D_{x\to j}-\overline{D}_x\right)}{\sqrt{\sum_{j\in\mathbb{I}}\left(D_{i\to j}-\overline{D}_i\right)^2}\sqrt{\sum_{j\in\mathbb{X}}\left(D_{x\to j}-\overline{D}_x\right)^2}}$$

where $\mathbb{I}$ and $\mathbb{X}$ are the set of workers that have interacted with the task publishers i and x, respectively. $\mathbb{C}=\mathbb{I}\cap\mathbb{X}$ is the set of workers that both have interacted with the task publisher i and task publisher x. $\overline{D}_i$ and $\overline{D}_x$ are the average values of direct reputation opinions for their interacted workers in $\mathbb{C}$, respectively. $D_{i\to j}$ and $D_{x\to j}$ are the reputation opinions for the worker j from task publishers i and x, respectively. A larger similarity factor represents that the reputation opinions from the recommender, i.e., task publisher x, are more trusted. Hence, the overall weight of indirect reputation opinions from recommender x is denoted as $\varpi_{i\to x}=\delta_{i\to x}\times\mathrm{Sim}(i,x)$, where $0\le\delta_{i\to x}\le1$ is a predefined parameter on behalf of the weight of recommended opinions from x for the task publisher i during calculation.

All the indirect reputation opinions from recommenders for the worker j can be integrated into an overall recommended reputation opinion in the form of $\Upsilon_{x\to j}^{\mathrm{rec}}:=\{b_{x\to j}^{\mathrm{rec}}, d_{x\to j}^{\mathrm{rec}}, u_{x\to j}^{\mathrm{rec}}\}$. We can obtain

$$\begin{cases} b_{x\to j}^{\mathrm{rec}}=\frac{1}{\sum_{x\in X}\varpi_{i\to x}}\sum_{x\in X}\varpi_{i\to x}b_{x\to j} \\ d_{x\to j}^{\mathrm{rec}}=\frac{1}{\sum_{x\in X}\varpi_{i\to x}}\sum_{x\in X}\varpi_{i\to x}d_{x\to j} \\ u_{x\to j}^{\mathrm{rec}}=\frac{1}{\sum_{x\in X}\varpi_{i\to x}}\sum_{x\in X}\varpi_{i\to x}u_{x\to j} \end{cases} \tag{8}$$

where $x\in X$ is the set of recommenders that had interacted with worker j. Hence, the indirect reputation opinions from different recommenders are combined into an overall recommended reputation opinion according to the corresponding weight $\varpi_{i\to x}$ of task publisher i [11].

C. Combining Direct Reputation Opinions With Recommended Reputation Opinions

When forming the compositive reputation value as the final reputation for the worker candidate j, the task publisher i considers not only the recommended reputation opinions, but also the direct reputation opinions to avoid cheating by other task publishers [10]. The final reputation opinion is denoted as $\Upsilon_{x\to j}^{\text{final}} := \{b_{x\to j}^{\text{final}}, d_{x\to j}^{\text{final}}, u_{x\to j}^{\text{final}}\}$, where

$$\begin{cases} b_{i\to j}^{\text{final}} = \frac{b_{i\to j}u_{x\to j}^{\text{rec}} + b_{x\to j}^{\text{rec}}u_{i\to j}}{u_{i\to j}+u_{x\to j}^{\text{rec}}-u_{x\to j}^{\text{rec}}u_{i\to j}} \\ d_{i\to j}^{\text{final}} = \frac{d_{i\to j}u_{x\to j}^{\text{rec}} + d_{x\to j}^{\text{rec}}u_{i\to j}}{u_{i\to j}+u_{x\to j}^{\text{rec}}-u_{x\to j}^{\text{rec}}u_{i\to j}} \\ u_{i\to j}^{\text{final}} = \frac{u_{x\to j}^{\text{rec}}u_{i\to j}}{u_{i\to j}+u_{x\to j}^{\text{rec}}-u_{x\to j}^{\text{rec}}u_{i\to j}}. \end{cases} \tag{9}$$

Hence, similar to (5), the final reputation value of task publisher i for the worker candidate j is $T_{i\to j}^{\text{final}} = b_{i\to j}^{\text{final}} + a u_{i\to j}^{\text{final}}$.

After reputation calculation in the above steps, the task publishers can select high-reputation worker candidates with high-accuracy and reliable data as the workers for model training in federated learning tasks. As mentioned in step 5 presented in Section IV, the task publishers also upload their reputation opinions for worker candidates to the reputation blockchain, which will work as the recommended reputation opinions for other task publishers. This reputation calculation scheme can recognize malicious worker candidates and avoid collusion cheating by their compromised task publishers [10]. More details are given in Section VII.

VI. Incentive Mechanism for Reliable Federated Learning

For reliable federated learning, we design an incentive mechanism to motivate high-reputation workers with high-quality local data to join model training. Note that high-quality data means high-accuracy and reliable local training data. Each task publisher offers a reward to the high-reputation workers that make contributions to federated learning tasks. However, there exist the following information asymmetry issues for the task publishers to do so [24], [25].

1) A task publisher does not know which mobile devices would like to join the model training due to the lack of prior knowledge.
2) The accurate reputation value and the local data quality of a worker are unknown to the task publisher.
3) The task publisher does not know the amount of available computation resource and the data sizes from workers for model training. As a result, the task publisher may suffer from too much cost when providing incentives to the mobile devices. It is essential for the task publishers to design an efficient incentive mechanism for reducing the impact of information asymmetry. Moreover, the workers with more contributions should be rewarded more. Therefore, contract theory as an efficient and powerful incentive mechanism is employed in this section [9], [25]–[27].

A. Computation Model for Federated Learning

We consider a federated learning task as a monopoly market with a monopolist operator (a task publisher) and a set of mobile devices $\mathcal{N} = \{1, \ldots, N\}$. Each worker $n \in \mathcal{N}$ with a local training dataset uses a size s_n of its local data samples to participate in the federated learning task. There is an input–output pair in each data sample, in which the input is a sample vector with various data features and the output is the label value for the input generated through mobile apps [3]. The contributed computation resources for local model training, i.e., CPU cycle frequency, from the worker n is denoted as f_n. The number of CPU cycles for a worker n to perform one sample of data[1] in local model training is denoted by c_n. Hence, for worker n, the computation time of a local iteration in local model training is $[c_n s_n/f_n]$. According to [3], the CPU energy consumption of the worker for one local iteration is expressed as follows:

$$E_n^{cmp}(f_n) = \zeta c_n s_n f_n^2 \tag{10}$$

where ζ is the effective capacitance parameter of computing chipset for worker n [26].

B. Communication Model for Federated Learning

For a federated learning task, all the workers (i.e., high-reputation mobile devices with high-quality data) collaborate to train a shared global model and achieve a global accuracy level of learning by an iterative method with a number of communication rounds (i.e., global iterations). During a global iteration, the workers send their own local model updates to the task publisher through wireless communications. Each local model update from worker n is affected by its local data quality, which is denoted as ε_n. The local data quality ε_n mainly depends on local data accuracy and data reliability, and can be normalized to a range. Note that, more accurate or reliable data brings larger ε_n. Intuitively, a better data quality (i.e., larger value of ε_n) leads to fewer local and global iterations and also improves the accuracy of training models. For ease of analysis, we use $\log(\frac{1}{\varepsilon_n})$ to represent the number of iterations of a local model update when the global accuracy is fixed [3], [28], which can be easily extended to more complicated expressions. There are computation time of a local iteration and uplink communication time[2] of a local model update during a global iteration. The computation time of a local iteration of the worker n is denoted by $T_n^{cmp} = [c_n s_n/f_n]$. For the communication time of local model updates, time-sharing multiaccess protocols, e.g., time-division medium access (TDMA) technology, can be taken into consideration in this article. We consider that the locations of workers are fixed during transmitting the local model parameters. The transmission rate of worker n is denoted as $r_n = B\ln(1 + [\rho_n h_n/N_0])$. Here, B is the transmission bandwidth and ρ_n is the transmission power of the worker n. h_n is the channel gain of peer-to-peer link between the worker n and the task publisher. N_0 is the background

[1] We consider that all the data samples have the same size.

[2] We consider that the downlink time between the task publisher and the workers is negligible compared with the uplink time as typically the downlink bandwidth is much larger than the uplink bandwidth.

noise. We consider the data size of a local model update is σ that is a constant with the same value for all workers. The transmission time of a local model update with the data size of σ is expressed by $T_n^{\text{com}} = (\sigma/[B\ln(1+[\rho_n h_n/N_0])])$.

Therefore, the total time of one global iteration for the worker n is denoted as

$$T_n^t = \log\left(\frac{1}{\varepsilon_n}\right) T_n^{cmp} + T_n^{\text{com}}. \tag{11}$$

According to [3], the energy consumption of the worker n to transmit local model updates in a global iteration is $E_n^{\text{com}} = T_n^{\text{com}} \cdot \rho_n = (\sigma\rho_n/[B\ln(1+[\rho_n h_n/N_0])])$.

Therefore, for a global iteration, the total energy consumption of the worker n is denoted as follows:

$$E_n^t = \log\left(\frac{1}{\varepsilon_n}\right) E_n^{cmp} + E_n^{\text{com}}. \tag{12}$$

C. Profit Function of the Task Publisher

To attract more workers with high-quality data (i.e., high-accuracy and reliable local data), we define a parameter about data quality as the type of the workers, which is denoted as

$$\theta_n = \frac{\psi}{\log\left(\frac{1}{\varepsilon_n}\right)}. \tag{13}$$

Here, ψ is the coefficient about the number of local model iterations affected by the local data accuracy. The workers are divided into N types, which are sorted in an ascending order of data quality: $\theta_1 < \cdots < \theta_n < \cdots < \theta_N, n \in \{1,\ldots,N\}$. The larger θ_n means the better quality of local data with higher accuracy, which brings fewer local model iterations [9], [18]. Although the task publisher does not know exactly true type of a given worker, it has the knowledge of the probability that a worker belongs to a certain type-n [29], and $\sum_{n=1}^{N} p_n = 1$. The task publisher can obtain the distribution of worker types from observations and statistics of previous behaviors of the workers [26], [27].

Due to information asymmetry, the task publisher should design specific contracts for different types of workers with different levels of data quality to increase its profits. The task publisher offers different resource-reward bundles to the workers according to their types. For different workers with different computation resources, i.e., CPU cycle frequency, the task publisher provides the contract $(R_n(f_n), f_n)$ including a series of resource-reward bundles to the workers. Here, f_n is the computation resource of type-n worker and $R_n(f_n)$ is the corresponding reward for the worker. The more computation resource contributed by the worker leads to faster local model training, thus bringing higher reward. The workers choose and sign one of the provided contracts at will and finish the given federated learning task. If the workers cannot finish the learning task or misbehave, the task publisher treats this interaction event as a negative event for reputation calculation in (4) and withhold payment.

For a signed contract $(R_n(f_n), f_n)$, we define the profit of the task publisher obtained from a type-n worker as follows:

$$U_{TP}(R_n) = \omega\ln\left(T_{\max} - T_n^t\right) - lR_n \tag{14}$$

where $\omega > 0$ is the satisfaction degree parameter of task publisher. $T_{\max}$ is the task publisher's maximum tolerance time of federated learning, and l is the unit cost about the rewards for the workers. Without loss of generality, $[\omega\ln(T_{\max} - T_n^t)]$ is denoted as the satisfaction function of the task publisher regarding the total time of one global iteration for type-n worker, which can also be extended to more sophisticated expressions [6]. The satisfaction of the task publisher will increase when the total time of one global iteration decreases. Note that both the better data quality (higher type) and larger CPU cycle frequency can decrease the global iteration time thus improving the profit of the task publisher, i.e., $[\partial U_{TP}/\partial\varepsilon_n] > 0$, $[\partial U_{TP}/\partial\theta_n] > 0$, and $[\partial U_{TP}/\partial f_n] > 0$. Moreover, for the task publisher, the more high-type workers joining the federated learning leads to more profit, but also causes larger reward cost lR_n. Apparently, the task publisher will not accept a negative profit when performing the federated learning task, i.e., $U_{TP}(R_n) \geq 0$. The objective of the task publisher is to maximize its profit in the federated learning task defined as follows:

$$\max_{(\mathrm{R_n},\mathrm{f_n})} U_{TP} = \sum_{n=1}^{N} Np_n \cdot w\ln\left[T_{\max} - \left(\frac{\sigma}{B\ln\left(1+\frac{p_n h_n}{N_0}\right)} + \frac{\psi}{\theta_n}\cdot\frac{c_n s_n}{f_n}\right)\right] - lR_n. \tag{15}$$

D. Utility Function of Workers

The utility function of a type-n worker for the signed contract $(R_n(f_n), f_n)$ is defined as

$$U_D(f_n) = R_n - \mu E_n^t = R_n - \mu\left[\frac{\psi}{\theta_n}\zeta c_n s_n f_n^2 + \frac{\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_0}\right)}\right] \tag{16}$$

where μ is a predefined weight parameter for energy consumption. We consider that every worker is self-interested and the valuation of U_D is zero when there is no reward [29]. Intuitively, the higher-type workers have larger utility since they provide better quality data. The types of the workers are determined by their local data quality (e.g., data accuracy and reliability). The workers with high-quality local data can obtain larger utility. Poisoning attacks launching by the malicious workers degrade their local data accuracy thus affecting the quality of local model updates. With the help of contract theory, the malicious workers will not choose to sign the contract items higher than their types due to individual rationality (IR) (more details about the IR are provided in Section VI-E). Therefore, the utility function of the workers based on contract theory can incentivize high-quality workers without adverse behavior to join the federated learning.

The worker wishes to minimize energy consumption when performing the federated learning task for maximizing its utility. The overall goal of a type-n worker is expressed by

$$\max_{(R_n,f_n)} U_D = R_n - \mu\left[\frac{\psi}{\theta_n}\zeta c_n s_n f_n^2 + \frac{\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_0}\right)}\right]. \tag{17}$$

E. Optimal Contract Design

In the case of information asymmetry, to make contracts feasible, each contract must satisfy the following constraints: 1) IR and 2) incentive compatibility (IC) in order to ensure that each type of workers are fully motivated [29].

Definition 1 (Individual Rationality): Every worker only participates in the federated learning task when the utility of the worker is not less than zero, i.e.,

$$U_D = R_n - \mu\left[\frac{\psi}{\theta_n}\zeta c_n s_n f_n^2 + \frac{\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_0}\right)}\right] \geq 0. \quad (18)$$

Definition 2 (Incentive Compatibility): To maximize utility, every worker can only choose the contract designed for itself, i.e., type θ_n, instead of any other contracts (R_m, f_m), i.e.,

$$\begin{aligned} R_n - \mu\left[\frac{\psi}{\theta_n}\zeta c_n s_n f_n^2 + \frac{\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_0}\right)}\right] \\ \geq R_m - \mu\left[\frac{\psi}{\theta_m}\zeta c_n s_n f_n^2 + \frac{\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_0}\right)}\right] \\ \forall n, m \in \{1,\ldots,N\}, n \neq m. \end{aligned} \quad (19)$$

In what follows, for simplicity, we consider $\mu = 1$. Without loss of generality, we consider the transmission bandwidth, transmission power, and the channel gain of all the workers are identical due to similar wireless communication environment [3], and thus we have $E_1^{\text{com}} = \cdots = E_n^{\text{com}} = (\sigma\rho_0/[B\ln(1+[\rho_0 h_0/N_0])]), n \in \{1,\ldots,N\}$. For ease of presentation, the optimization problems in (15) and (17) can be reformulated as

$$\begin{aligned} \max_{(R_n,f_n)} \quad & U_{TP} = \sum_{n=1}^{N} N p_n\left[w\ln\left(T_{\max} - T_n^{\text{com}} - \frac{\psi T_n^{cmp}}{\theta_n}\right) - lR_n\right] \\ \text{s.t.} \quad & R_n - \left(\frac{\psi}{\theta_n}E_n^{cmp} + E_n^{\text{com}}\right) \geq 0, \forall n \in \{1,\ldots,N\} \\ & R_n - \left(\frac{\psi}{\theta_n}E_n^{cmp} + E_n^{\text{com}}\right) \geq R_m - \left(\frac{\psi}{\theta_n}E_m^{cmp} + E_m^{\text{com}}\right) \\ & \forall n, m \in \{1,\ldots,N\}, n \neq m \\ & \frac{c_n s_n}{f_n} \leq T_{\max}, \forall n \in \{1,\ldots,N\} \\ & \sum_{n=1}^{N} N \cdot p_n \cdot R_n \leq R_{\max}, \forall n \in \{1,\ldots,N\} \end{aligned} \quad (20)$$

where $R_{\max}$ is the total amount of the given reward from the task publisher. Although the problem in (20) is not a convex optimization problem, its solution can be found by performing the following transformation.

According to the above definitions, we have the following lemmas.

Lemma 1 (Monotonicity): For contract (R_n, f_n) and (R_m, f_m), we have $f_n \geq f_m$ and $R_n \geq R_m$, if and only if $\theta_n \geq \theta_m$, $n \neq m$, and $n, m \in \{1,\ldots,N\}$.

Proof: Based on the IC constraints of type-n workers and type-m workers, we have

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} \geq R_m - \frac{\psi}{\theta_n}E_m^{cmp} \quad (21)$$

$$R_m - \frac{\psi}{\theta_n}E_m^{cmp} \geq R_n - \frac{\psi}{\theta_n}E_n^{cmp}. \quad (22)$$

By summing up the above inequalities (21) and (22) together, we have $([\psi/\theta_n] - [\psi/\theta_n])(E_n^{cmp} - E_m^{cmp}) \geq 0$. Note that $E_n^{cmp}(f_n)$ is a monotonically increasing valuation function with respect to f_n. It is easy to find that $(E_n^{cmp} - E_m^{cmp}) \geq 0$, i.e., $f_n \geq f_m$ holds if $\theta_n \geq \theta_m$. When $f_n \geq f_m$, we have $(E_n^{cmp} - E_m^{cmp}) \geq 0$, then $\theta_n \geq \theta_m$ must be satisfied [9]. ■

Proposition 1: $R_n \geq R_m$ if and only if $f_n \geq f_m$.

Proof: According to the IC constraints in (21) and (22), we have the following inequalities:

$$R_n - R_m \geq \frac{\psi}{\theta_n}(E_n^{cmp} - E_m^{cmp}) \quad (23)$$

$$R_n - R_m \leq \frac{\psi}{\theta_n}(E_n^{cmp} - E_m^{cmp}). \quad (24)$$

When $f_n \geq f_m$, we have $E_n^{cmp} \geq E_m^{cmp}$. Thus, we can deduce $R_n \geq R_m$ from (23). In addition, as $R_n \geq R_m$, we can deduce $E_n^{cmp} \geq E_m^{cmp}$, i.e., $f_n \geq f_m$, from (24). According to Proposition 1, we know that an IC contract requires a higher reward when the workers contribute faster CPU-cycle frequency for federated learning tasks. ■

Lemma 2: If the IR constraint of type-1 is satisfied, the other IR constraints will also hold.

Proof: According to the IC constraints, $\forall n \in \{2,\ldots,N\}$, we can obtain

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} \geq R_1 - \frac{\psi}{\theta_n}E_1^{cmp}. \quad (25)$$

Given that $\theta_1 < \cdots < \theta_n < \cdots < \theta_N$, we can also obtain

$$R_1 - \frac{\psi}{\theta_n}E_1^{cmp} \geq R_1 - \frac{\psi}{\theta_1}E_1^{cmp}. \quad (26)$$

Combing (25) and (26), we have

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} \geq R_1 - \frac{\psi}{\theta_1}E_1^{cmp}.$$

Since all the energy consumption of transmitting local model for workers are the same, we have

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} - E_n^{\text{com}} \geq R_1 - \frac{\psi}{\theta_1}E_1^{cmp} - E_1^{\text{com}}. \quad (27)$$

Equation (27) implies that when the IR constraint of type-*1* worker is satisfied, the other IR constraints will automatically hold. Hence, the other IR constraints can be bounded into the IR condition of type-*1* worker [9], [30]. Note that the (28) is tight for the optimization problem in (21). ■

Lemma 3: According to the monotonicity in Lemma 1, the IC condition can be reduced as the local downward incentive constraints (LDIC) that is expressed as follows:

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} \geq R_{n-1} - \frac{\psi}{\theta_n}E_{n-1}^{cmp}, \forall n \in \{2,\ldots,N\}. \quad (28)$$

Proof: The IC constraints between type-n and type-m, $\forall m \in \{1, \ldots, n-1\}$ are defined as downward IC (DIC), which is expressed as $R_n - [\psi/\theta_n]E_n^{cmp} \geq R_m - [\psi/\theta_n]E_m^{cmp}$.

The IC constraints between type-n and type-m, $\forall m \in \{n+1, \ldots, N\}$ are defined as upward IC (UIC), which is expressed as $R_n - [\psi/\theta_n]E_n^{cmp} \geq R_m - [\psi/\theta_n]E_m^{cmp}$.

We first prove that DIC can be reduced as two adjacent types in DIC, called LDIC. Given $\theta_{n-1} < \theta_n < \theta_{n+1}$, $n \in \{2, \ldots, N-1\}$, we can obtain

$$R_{n+1} - \frac{\psi}{\theta_{n+1}}E_{n+1}^{cmp} \geq R_n - \frac{\psi}{\theta_{n+1}}E_n^{cmp} \tag{29}$$

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} \geq R_{n-1} - \frac{\psi}{\theta_n}E_{n-1}^{cmp}. \tag{30}$$

By utilizing the monotonicity, i.e., $R_n \leq R_m$ if and only if $\theta_n \leq \theta_m$, $n \neq m$, and $n, m \in \{1, \ldots, N\}$, we can obtain

$$\theta_{n+1}(R_n - R_{n-1}) \geq \theta_n(R_n - R_{n-1}). \tag{31}$$

Combing (30) and (31), we have

$$R_n - \frac{\psi}{\theta_{n+1}}E_n^{cmp} \geq R_{n-1} - \frac{\psi}{\theta_{n+1}}E_{n-1}^{cmp}. \tag{32}$$

Combing (29) and (32), we have

$$R_{n+1} - \frac{\psi}{\theta_{n+1}}E_{n+1}^{cmp} \geq R_{n-1} - \frac{\psi}{\theta_{n+1}}E_{n-1}^{cmp}. \tag{33}$$

Equation (33) can be extended to prove that all the DIC can be held until type-1

$$R_{n+1} - \frac{\psi}{\theta_{n+1}}E_{n+1}^{cmp} \geq R_{n-1} - \frac{\psi}{\theta_{n+1}}E_{n-1}^{cmp} \geq \cdots \geq R_1 - \frac{\psi}{\theta_{n+1}}E_1^{cmp}, \forall n \in \{1, \ldots, N-1\}. \tag{34}$$

Hence, with the LDIC, all the DICs hold and can be reduced. Similarly, we can prove that by using the monotonicity, given the local UIC (LUIC) holds, all the UICs can automatically hold [9], [30]. ∎

Based on the analysis in Lemmas 1–3 and the LDIC is tight at the optimal point, the optimization problem in (20) is simplified as follows:

$$\max_{(\mathrm{R_n},\mathrm{f_n})} U_{TP} = \sum_{n=1}^{N} Np_n\left[w\ln\left(T_{\max} - T_n^{\mathrm{com}} - \frac{\psi T_n^{cmp}}{\theta_n}\right) - lR_n\right]$$

s.t.

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} - E_n^{\mathrm{com}} = 0, \forall n \in \{1, \ldots, N\}$$

$$R_n - \frac{\psi}{\theta_n}E_n^{cmp} = R_{n-1} - \frac{\psi}{\theta_n}E_{n-1}^{cmp}, \forall n \in \{2, \ldots, N\}$$

$$\frac{c_n s_n}{f_n} \leq T_{\max}, \forall n \in \{1, \ldots, N\}$$

$$\sum_{n=1}^{N} N \cdot p_n \cdot R_n \leq R_{\max}, \forall n \in \{1, \ldots, N\}. \tag{35}$$

To derive the optimal contracts in the problem (35), we first solve the relaxed problem in (35) without monotonicity constraint. Subsequently, this acquired solution is checked whether it satisfies the monotonicity condition. By using the iterative method on *IC* and *IR* constraints, we can obtain the reward which is expressed as

$$R_n = E_n^{\mathrm{com}} + \frac{\psi E_1^{cmp}}{\theta_1} + \sum_{k=1}^{n}\Delta_k \tag{36}$$

where $\Delta_k = [\psi E_k^{cmp}/\theta_k] - [\psi E_{k-1}^{cmp}/\theta_k]$ and $\Delta_1 = 0$. By substituting R_n into $\sum_{n=1}^{N} N \cdot p_n \cdot lR_n$, we can obtain

$$\sum_{n=1}^{N} N \cdot p_n \cdot lR_n = NlE_n^{\mathrm{com}} + Nl\zeta\sum_{n=1}^{N} g_n c_n s_n f_n^2 \tag{37}$$

where

$$g_n = \begin{cases} \frac{\psi P_n}{\theta_n} + \left(\frac{\psi}{\theta_n} - \frac{\psi}{\theta_{n+1}}\right)\sum_{i=n+1}^{N} p_i, & n < N \\ \frac{\psi P_N}{\theta_N}, & n = N. \end{cases}$$

By substituting (37) into the problem in (35) and also removing all R_n, we can rewrite (35) as

$$\max_{(R_n, f_n)} U_{TP} = \sum_{n=1}^{N} Np_n\left[w\ln\left(T_{\max} - \frac{\sigma}{B\ln\left(1+\frac{\rho_n h_n}{N_n}\right)} - \frac{\psi c_n s_n}{f_n\theta_n}\right)\right] - \frac{Nl\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_n}\right)} - Nl\zeta\sum_{n=1}^{N} c_n s_n g_n f_n^2$$

s.t.

$$\frac{c_n s_n}{T_{\max}} \leq f_n, \forall n \in \{1, \ldots, N\}$$

$$\frac{N\sigma\rho_n}{B\ln\left(1+\frac{\rho_n h_n}{N_n}\right)} + N\zeta\sum_{n=1}^{N} c_n s_n g_n f_n^2 \leq R_{\max}, \forall n \in \{1, \ldots, N\}. \tag{38}$$

By differentiating U_{TP} with respect to f_n, we can obtain $[\partial^2 U_{TP}/\partial f_n^2] < 0$, and thus U_{TP} is concave. The summation of concave functions (U_{TP}) is still a concave function, and hence the problem in (38) with affine constraints is a concave optimization problem. With the help of convex optimization tools, e.g., CVX, we can calculate the optimal computation resource, i.e., contributed CPU-cycle f_n^* and the corresponding incentive R_n^* [9]. In addition, the monotonicity can be automatically met when the types of workers follow uniform distribution [31]. If the distribution of the workers' type is not uniform, we can utilize the infeasible subsequence replacing algorithm to meet the final optimal computation resource requirement [30], [32].

VII. Performance Evaluation

A. Simulation Setting

In the simulation, we utilize a well-known digit classification dataset named MNIST and a widely used software environment TensorFlow 1.12.0 to perform a digit classification task to evaluate the proposed incentive schemes. There are 60 000 training examples and 10 000 test examples in the MNIST dataset, which is widely adopted in machine learning evaluation, e.g., federated learning [19], [26]. There exist 20 task publishers, 90 well-behaved workers, and ten unreliable workers in the federated learning tasks. These unreliable workers include eight workers with unintentionally unreliable local

TABLE II
PARAMETER SETTING IN THE SIMULATION

Parameter	Setting
Interaction frequency between a task publisher and workers	[15, 30] federated learning tasks a week
Weight parameters	$\kappa = 0.4$, $\eta = 0.6$
Successful transmission probability of data packets	[0.8, 1]
Local data quality ε_n	[0.2, 0.92]
CPU cycles for performing a data sample	$c_n = c_0 = 5$
The size of data samples	$s_n = s_0 = 20$
The transmission time and energy consumption of transmitting a local model update	$T_n^{com} = 10$, $E_n^{com} = 20, n \in \{1, \ldots, N\}$
The maximum tolerance time of a federated learning task	$T_{max} = 600$
The total amount of given reward	$R_{max} = 10000$
Pre-defined parameters	$a = 0.5$, $z = 0.8$, $\delta_{i \to x} = 0.9$, $\zeta = 2$, $\psi = 15$, $w = 800$, $\mu = 1$, $l = 1$

update caused by their low-quality data and two malicious workers launching poisoning attacks. Specifically, each well-behaved worker is randomly assigned a training set following a uniform distribution over ten classes as its local training data, while every unreliable worker only receives a certain number of classes randomly. To quantify the data quality of local training data in the unreliable workers, we use the earth mover's distance (EMD) as an index. The EMD is denoted by the probability distance for a worker's training data distribution compared with the actual distribution for the whole population [33]. For the malicious workers with poisoning attacks, they randomly have training data with ten classes. Nevertheless, some labels of the training examples are deliberately modified to mislead the global model training. The modified label percentage of the training examples indicates the attack strength of a poisoning attack. Each model is trained for five synchronous iterations among workers, and the workers use a batch of 32 randomly sampled training examples to generate a local SGD update [19]. For simplicity, we set all the transmission rates of local model updates as the same value, thus the transmission time and energy consumption for all the workers are the same.

For reputation calculation, we consider that the task publishers consist of 15 honest task publishers and five compromised task publishers. The compromised task publishers provide fake recommended reputation opinions to other task publishers for misleading reputation calculation. We compare the proposed multiweight subjective logic scheme with a TSL scheme presented in [9] and [11]. The initial reputation of the unreliable workers is 0.49. A reputation blockchain is established on Corda V4.0 that provides reliable transaction services [34]. We employ PBFT as the consensus algorithm running on 30 preselected miners. SHA-256 is used as the secure hash algorithm and the data size of a reputation opinion is from 200B to 400B. The block size limit of reputation opinions is 1 MB. The average block generation time is 10 s [35]. For the incentive mechanism, the worker candidates are initially classified into ten types according to accuracy-related parameters of local training data, and the probability for a candidate belonging to a certain type is 0.1 [9]. More parameters used in the simulation are given in Table II [9], [11], [30].

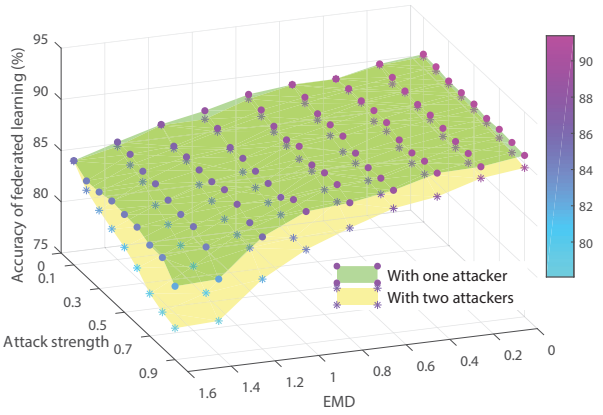


Fig. 2. Accuracy of federated learning tasks under different data quality levels and attack strengths.

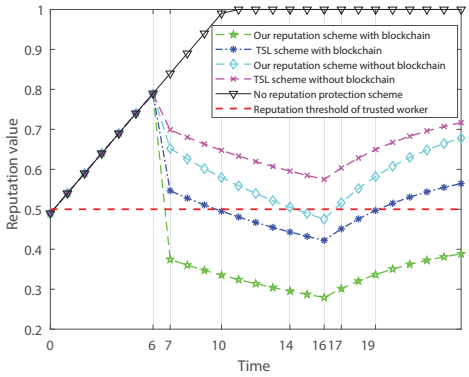


Fig. 3. Reputation values of an unreliable worker.

B. Performances of the Proposed Reputation Scheme

We first evaluate the effects of the unreliable workers on accuracy of federated learning tasks with different levels of data quality, i.e., EMDs and poisoning attack strengths. From Fig. 2, we can observe that the increasing EMD, larger attack strength, or more attackers can bring decreasing federated learning accuracy. Consequently, both the unreliable workers with low-quality training data (i.e., large EMD) and attackers with poisoning attack cause negative effects on the learning accuracy. For example, when the EMD is 1.6 and the attack strength is 0.9, the accuracy of federated learning with two attackers is 78.07%, which is 5% lower than that with one attackers. When the attack strength is 0.9 and the EMD is 1.6, the accuracy is 6.6% lower than that with zero EMD in the case of two attackers.

To illustrate the performance of different reputation calculation schemes with and without reputation blockchain, we consider that an unreliable worker intentionally well behaves to improve its reputation value in the former six federated tasks. Thereafter, this worker randomly misbehaves to 15 task publishers with the probability of 0.8, which trains local models on local data with poisoning or unreliable examples. For the schemes without the reputation blockchain, we consider that all reputation opinions are centralizedly stored in a cloud server. However, there exists an internal attack in the

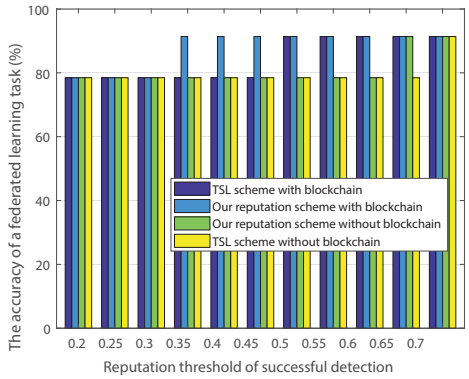


Fig. 4. Accuracy under different reputation thresholds of successful detection (EMD = 1.6, attack strength = 0.8).

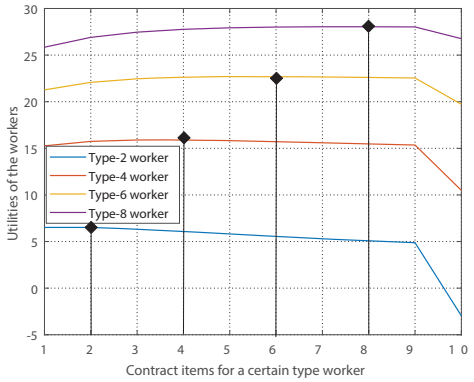


Fig. 5. Utilities of workers with different contract items.

central cloud that an attacker may manipulate the stored reputation opinions for the unreliable worker into good reputation opinions with a probability of 50%.

Fig. 3 shows that, when performing misbehaviors, the reputation of the unreliable worker in all the schemes with reputation begins to decrease. However, for the baseline scheme without reputation protection, the reputation is still linearly increasing. The reputation in the proposed reputation calculation schemes with & without the reputation blockchain has a sharper and larger decrease than those in the TSL schemes with & without the reputation blockchain in a short time period because of the weight effects of interaction effects and freshness. Moreover, when the unreliable worker pretends to behave well again, its reputation in TSL schemes with & without the blockchain increases faster than those in the proposed reputation calculation schemes with & without the reputation blockchain. After four more learning tasks, the worker becomes trusted in TSL scheme with the blockchain when the reputation threshold of trusted worker is 0.5, while the worker is still untrusted in the proposed reputation calculation scheme with the blockchain. Here, the reputation threshold of the trusted worker represents that a worker candidate with higher reputation than a given threshold is treated as a trustable candidate in the reputation calculation schemes. Moreover, compared with the schemes without the reputation blockchain, the reputation in the both schemes with the blockchain is more accurate and reliable because of the reputation blockchain with the nature properties of tramper-resistance and decentralization that can defend against the internal attack.

To show the effects of reputation threshold on federated learning accuracy, we evaluate the accuracy of a federated learning task with respect to different reputation thresholds of successful detection and different reputation schemes. Here, the reputation threshold of successful detection means that a maliciously unreliable worker with the calculated reputation value from different schemes below the given reputation threshold is detected successfully. Fig. 4 shows that the higher reputation threshold leads to a higher training accuracy. For the schemes with the reputation blockchain, the accuracy of the proposed reputation scheme is higher than that of the TSL scheme when the reputation threshold ranges from 0.35 to 0.45. By contrast, for the schemes without the blockchain, the accuracy of the federated learning tasks is not lower than those with the blockchain until the reputation threshold of successful detection being higher than 0.65. The reason is that, with the help of the reputation blockchain, the reputation opinions from the honest task publishers can be securely stored and protected without manipulation, leading to more accurate reputation calculation and more reliable worker selection for better learning performance. Nevertheless, all the schemes cannot completely detect the unreliable workers when the reputation threshold is relatively low, such as lower than 0.3. The reason is that the unreliable workers can disguise themselves well through good behaviors during some federated learning tasks, and thus they cannot be detected in a short time.

C. Performances of the Contract-Based Incentive Mechanism

To validate the feasibility, i.e., IR and IC, of the proposed scheme under information asymmetry, we present Fig. 5 to show the utilities of workers with types 2, 4, 6, and 8, respectively [30]. From Fig. 5, we observe that all types of workers can only achieve their own maximum utility when they choose the contract item exactly designed for their types, which explains the IC constraint [9]. Moreover, each worker can obtain non-negative utility when selecting the contract item corresponding to its type, which validates the IR constraint.

We compare the profit of the task publisher obtained from the proposed contract theory model, and that from the Stackelberg game model in [30]. Fig. 6 shows that a larger total number of worker types leads to the larger profit of a task publisher. The more worker types bring more contract item choices to high-type (high-reputation) workers, thus ensuring more reliable federated learning. For a certain number of worker types, the profit of the task publisher in the proposed contract model is higher than that of the Stackelberg game model. The reason is that, in the monopoly market, the task publisher working as the monopolist only provides limited contract items to the workers and extracts more profit from the workers. Nevertheless, in the Stackelberg game model, rational workers can optimize their individual utilities resulting in less profit for the task publisher. Although the task publisher needs to consider the IR and IC constraints during designing the contract items, these constraints have a small impact on maximizing the

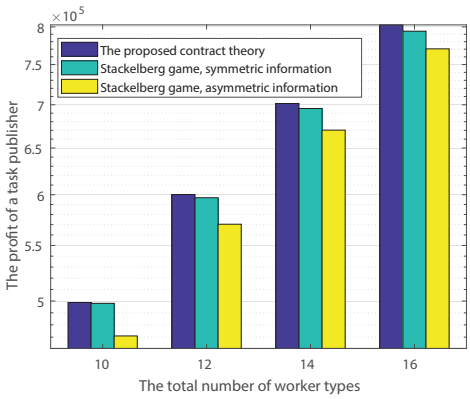


Fig. 6. Profit of a task publisher under different total number of worker types.

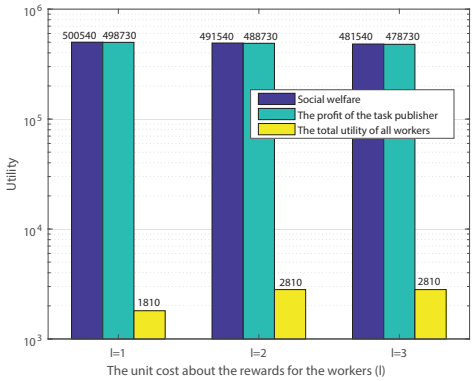


Fig. 7. Utility under different unit costs about rewards.

utilities of the workers compared with the Stackelberg game model [36]. As a result, the task publisher can obtain the higher profit than that in the Stackelberg game models [9]. Moreover, the Stackelberg game model with symmetric information has better performance than that of Stackelberg game model with asymmetric information. The reason is that the game leader (the task publisher) in the Stackelberg game with symmetric information can optimize its profit because of knowing the actions of followers (workers), i.e., the symmetric information, and set the utilities of the followers to zero [30].

For the task publisher, there exists unit cost about rewards for workers during federated learning tasks. Fig. 7 shows that in the contract-based incentive scheme, both the profit of the task publisher and the social welfare decrease with the increase of unit cost about rewards. The total utility of all workers increases with the increase of unit cost about rewards because of receiving larger reward from the task publisher. To illustrate the impacts of the variation range of local training data accuracy on the profit of task publisher, we vary the upper limit of local data accuracy (i.e., a parameter related to the worker type) from 98% to 88%, and 78%, respectively. As shown in Fig. 8, the profit of the task publisher decreases with the decrease of the upper limit of local data accuracy. As reducing upper limit of the local data accuracy means that the number of high-reputation workers is decreasing. Therefore, the low-quality of local training data has a negative impact on the profit of the task publisher. In summary,

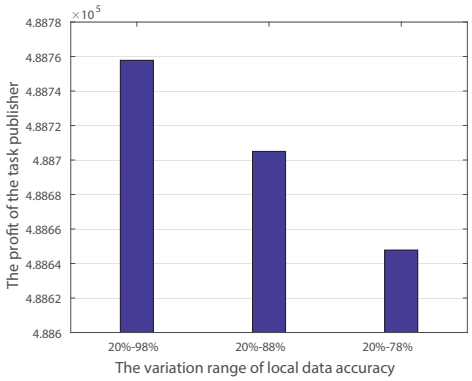


Fig. 8. Profit of the task publisher under different accuracy levels of local training data.

the proposed schemes achieve more accurate reputation calculation and stimulate high-reputation workers with accurate as well as reliable data to join learning tasks, hence leading to the more reliable federated learning.

VIII. Conclusion

In this article, we have focused on the worker selection and incentive mechanism issues for reliable federated learning in mobile networks. We have first introduced a reputation-based worker selection scheme for reliable worker selection. To ensure efficient and secure reputation management, we have employed a multiweight subjective logic model to calculate the reputation of the workers according to direct interaction histories and recommended reputation opinions, and thus utilized a consortium blockchain to manage the reputation of workers in a decentralized manner without repudiation. Furthermore, to enable reliable federated learning, we have designed contract theory-based incentive mechanism to stimulate high-reputation workers that have high-accuracy and reliable local training data to join the learning processes. Numerical results have confirmed that the contract-based incentive scheme can attract more high-reputation workers with high-quality local training data to ensure reliable federated learning and also optimize the utilities of both the task publishers and the workers.

In future work, to further improve the accuracy of reputation calculation, more weight parameters can be taken into consideration to optimize the reputation calculation. Motivated by the success of machine learning in security protection, we can utilize advanced machine learning techniques to detect and remove compromised task publishers or malicious workers to further enable reliable federated learning.

References

[1] 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. 1–15, 2019.

[2] X. Zhu, H. Li, and Y. Yu, "Blockchain-based privacy preserving deep learning," in *Proc. Int. Conf. Inf. Security Cryptol.*, 2018, pp. 370–383.

[3] N. H. Tran, W. Bao, A. Y. Zomaya, N. N. H. Minh, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1387–1395.

[4] M. Shayan *et al.* (2018). *Biscotti: A Ledger for Private and Secure Peer-to-Peer Machine Learning*. [Online]. Available: https://arxiv.org/abs/1811.09904

[5] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, to be published. doi: 10.1109/LCOMM.2019.2921755.

[6] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[7] Y. Li, C. Courcoubetis, and L. Duan, "Recommending paths: Follow or not follow?" in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 928–936.

[8] T. T. Anh, N. C. Luong, D. Niyato, D. I. Kim, and L.-C. Wang, "Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach," *IEEE Wireless Commun. Lett.*, to be published. doi: 10.1109/LWC.2019.2917133.

[9] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.

[10] Y. Liu, K. Li, Y. Zhang, and W. Qu, "A novel reputation computation model based on subjective logic for mobile ad hoc networks," *Future Gener. Comput. Syst.*, vol. 27, no. 5, pp. 547–554, 2011.

[11] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[12] 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.

[13] S. Delgado-Segura, C. Tanas, and J. Herrera-Joancomartí, "Reputation and reward: Two sides of the same bitcoin," *Sensors*, vol. 16, no. 6, pp. 1–23, 2016.

[14] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.

[15] A. Hard *et al.* (2018). *Federated Learning for Mobile Keyboard Prediction*. [Online]. Available: https://arxiv.org/abs/1811.03604

[16] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw. Mag.*, to be published.

[17] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency V2V communications," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2018, pp. 1–7.

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

[19] C. Fung *et al.* (2018). *Mitigating Sybils in Federated Learning Poisoning*. [Online]. Available: https://arxiv.org/abs/1808.04866

[20] H. Zhu and Y. Jin, "Multi-objective evolutionary federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[21] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, and D. I. Kim, "Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 157–160, Feb. 2019.

[22] M. Sohail, L. Wang, S. Jiang, S. Zaineldeen, and R. U. Ashraf, "Multi-hop interpersonal trust assessment in vehicular ad-hoc networks using three-valued subjective logic," *IET Inf. Security*, vol. 13, no. 3, pp. 223–230, May 2019.

[23] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[24] Y. Li, C. A. Courcoubetis, and L. Duan, "Dynamic routing for social information sharing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 571–585, Mar. 2017.

[25] J. Gao, L. Zhao, and X. Shen, "Network utility maximization based on an incentive mechanism for truthful reporting of local information," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7523–7537, Aug. 2018.

[26] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. 16th IEEE Asia–Pac. Wireless Commun. Symp.*, Singapore, Aug. 2019.

[27] M. Zeng, Y. Li, K. Zhang, M. Waqas, and D. Jin, "Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[28] J. Konečnỳ *et al.* (2016). *Federated Optimization: Distributed Machine Learning for On-Device Intelligence*. [Online]. Available: https://arxiv.org/abs/1610.02527

[29] Y. Zhang, L. Liu, Y. Gu, D. Niyato, M. Pan, and Z. Han, "Offloading in software defined network at edge with information asymmetry: A contract theoretical approach," *J. Signal Process. Syst.*, vol. 83, no. 2, pp. 241–253, 2016.

[30] Z. Hou, H. Chen, Y. Li, and B. Vucetic, "Incentive mechanism design for wireless energy harvesting-based Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2620–2632, Aug. 2018.

[31] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.

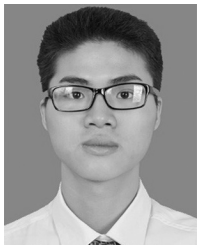
[32] L. Gao, X. Wang, Y. Xu, and Q. Zhang, "Spectrum trading in cognitive radio networks: A contract-theoretic modeling approach," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 843–855, Apr. 2011.

[33] Y. Zhao *et al.* (2018). *Federated Learning With Non-IID Data*. [Online]. Available: https://arxiv.org/abs/1806.00582

[34] J.-S. Weng, J. Weng, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," in *Proc. IACR Cryptol. ePrint Archive*, 2018, pp. 1–16.

[35] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.

[36] T. Liu, J. Li, F. Shu, M. Tao, W. Chen, and Z. Han, "Design of contract-based trading mechanism for a small-cell caching system," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6602–6617, Oct. 2017.



Jiawen Kang received the M.S. and Ph.D. degrees from the Guangdong University of Technology, Guangzhou, China, in 2015 and 2018, respectively.

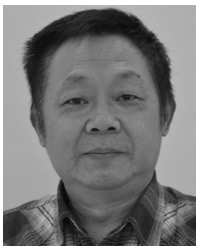
He is currently a Post-Doctoral Fellow with Nanyang Technological University, Singapore. His current research interests include blockchain, and security and privacy protection in wireless communications and networking.

Zehui Xiong (S'17) received the B.Eng. degree (Highest Hons.) in telecommunication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

He is a visiting Ph.D. student at Princeton University, Princeton, NJ, USA, in 2019. His current research interests include network economics, wireless communications, blockchain, and deep reinforcement learning.

Dusit Niyato (F'17) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include energy harvesting for wireless communication, Internet of Things, and sensor networks.



Shengli Xie (F'19) received the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1992, and the Ph.D. degree in automatic control from the South China University of Technology, Guangzhou, China, in 1997.

He was a Vice Dean of the School of Electronics and Information Engineering, South China University of Technology, from 2006 to 2010. He is currently the Director of both the Institute of Intelligent Information Processing and the Guangdong Key Laboratory of Information Technology for the Internet of Things, and also a Professor with the Guangdong University of Technology, Guangzhou. He has authored or coauthored four monographs and over 100 scientific papers published in journals and conference proceedings, and was granted over 30 patents. His current research interests include statistical signal processing and wireless communications, with an emphasis on blind signal processing and Internet of Things.

Junshan Zhang (F'12) received the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2000.

He joined the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA, in August 2000, where he has been the Fulton Chair Professor since 2015. His current research interests include the general field of information networks and data science, including communication networks, Internet of Things, fog computing, social networks, and smart grid.

Dr. Zhang was a recipient of the ONR Young Investigator Award in 2005, the NSF CAREER Award in 2003, and the IEEE Wireless Communication Technical Committee Recognition Award in 2016. His papers have won several awards, including the Kenneth C. Sevcik Outstanding Student Paper Award of ACM SIGMETRICS/IFIP Performance in 2016, the Best Paper Runner-Up Award of IEEE INFOCOM 2009 and IEEE INFOCOM 2014, and the Best Paper Award at IEEE ICC 2008 and ICC 2017. He was the TPC Co-Chair for a number of major conferences in communication networks, including IEEE INFOCOM 2012 and ACM MOBIHOC 2015. He was the General Chair of ACM/IEEE SEC 2017, WiOPT 2016, and IEEE Communication Theory Workshop 2007. He was a Distinguished Lecturer of the IEEE Communications Society. He was an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and an Editor of *Computer Networks* and the *IEEE Wireless Communication Magazine*. He is currently serving as the Editor-in-Chief for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Editor-at-Large for the IEEE/ACM TRANSACTIONS ON NETWORKING, and an Editor for IEEE NETWORK.