An Incentive Scheme for Federated Learning in the Sky

Wei Yang Bryan Lim[1], Zehui Xiong[1], Jiawen Kang[2], Dusit Niyato[2], Yang Zhang[3], Cyril Leung[1,4], and Chunyan Miao[1,2]

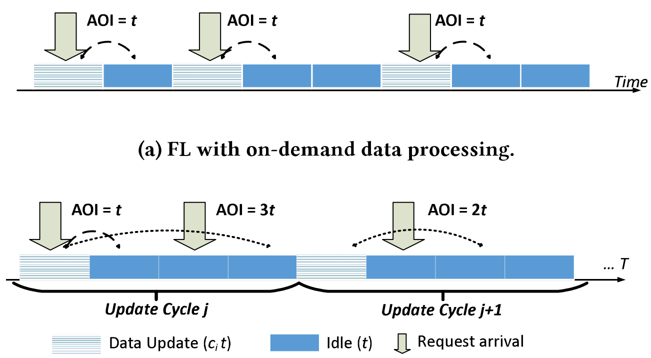
[1] Alibaba-Nanyang Technological University (NTU) Joint Research Institute, Singapore
[2] School of Computer Science and Engineering (SCSE), NTU, Singapore
[3] Hubei Key Laboratory of Transportation Internet of Things, School of Computer Science and Technology, Wuhan University of Technology, China
[4] University of British Columbia (UBC), Canada

ABSTRACT

The enhanced capabilities of Unmanned Aerial Vehicles have promoted the rapid growth of the Drones-as-a-Service (DaaS) market. To enable privacy-preserving collaborative machine learning among independent DaaS providers, we propose a Federated Learning (FL) based approach. There exists a tradeoff between Service Latency (SL), i.e., the time taken for the training request to be completed, and Age of Information (AoI), i.e., the time elapsed between data aggregation to completion of the FL based training. Given that different training tasks may have varying AoI requirements, we propose a contract-theoretic task-aware incentive scheme that can be calibrated based on the weighted preferences of the model owner. Performance evaluation validates the incentive compatibility and flexibility of our contract design amid information asymmetry.

KEYWORDS

Mobile crowdsensing, Unmanned Aerial Vehicle, Incentive Mechanism, Federated Learning



(a) FL with on-demand data processing.

(b) FL with caching.

Figure 1: AoI model of FL training.

1 INTRODUCTION

In recent years, the enhanced sensing and communication capabilities of UAVs have promoted the rapid growth of the DaaS market. The UAVs feature the benefits of high mobility, flexible deployment, cost effectiveness, and can provide a more comprehensive coverage for mobile crowdsensing sensing as compared to terrestrial devices [3].

However, the state-of-the-art representation-learning based models, e.g., Deep Learning (DL), typically require large quantities of training data to outperform conventional hand-crafted analytical methods. Naturally, independent DaaS providers, i.e., UAVs, can collaborate by sharing their data. However, the UAVs may be reluctant to share their private raw data with industrial competitors. Moreover, privacy concerns exist amid increasingly stringent regulations. As such, we propose the adoption of a FL [2] based approach to enable collaborative model training across a independent UAVs.

Our system model consists of UAVs, hereinafter *workers*, and a model owner, e.g., a city planner that employs the services of the UAVs. The worker first collects the data by flying over deployed wireless sensor networks for aggregation on its private edge server. When the model owner requires model training, a request and an initialized set of model parameters are transmitted to the worker. Then, the

DroneCom 20, September 25, 2020, London, United Kingdom

ACM ISBN 978-1-4503-8105-5/20/09…$15.00
https://doi.org/10.1145/3414045.3415935

worker trains the model on its local data and transmits only the updated parameters to the model owner for aggregation. The local training and transmission for global aggregation iterates till a desired accuracy is achieved.

In conventional FL studies, it is usually assumed that the worker either has the data ready for model training, or needs to collect the data upon request. As such, the crucial tradeoff between *SL* and *AoI* is under-explored. In particular, the AoI is defined as the time elapsed between data collection, i.e., collection or aggregation of data from distributed sensors, to completion of the FL based training, and thus captures information freshness [1]. In contrast, the SL is defined as the time elapsed between the initiation of the FL training request to the completion of the FL based training. On one hand, if the data is aggregated only upon request, the information is at its freshest, i.e., the AoI is low. On the other hand, the SL incurred during data collection can be intolerable especially if the model owner prefers faster task completion. Therefore, we propose a contract-theoretic task-aware incentive mechanism to motivate workers to update the data accordingly in consideration of the different preferences of the model owner towards AoI and SL.

2 PROBLEM FORMULATION

The model owner initiates a synchronous FL task involving a set $\mathcal{I} = \{1, \ldots, i, \ldots, I\}$ of I workers that lasts for a fixed duration T. During the FL task, there can be more than one instance of model training request initiated by the model owner, e.g., to ensure that the global model is kept up-to-date, through model training with updated data. We assume that each instance of request arrival follows the Poisson process [4]. An FL based model training is initiated through the request of the model owner. Each model training takes place over K iterations to minimize the global loss $F^K(\boldsymbol{w})$ where K is stipulated by the model owner and $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$. Each k^{th} iteration in turn consists of three steps [2] namely: (i) *Local Computation*: The worker trains the received global model $\boldsymbol{w}^{(k)}$ locally using the processed data, (ii) *Wireless Transmission*: The worker transmits the model parameter update to the model owner, and (iii) *Model Parameter Update*: All parameter updates received from the I workers are aggregated to derive an updated global model $\boldsymbol{w}^{(k+1)}$, which is then transmitted back to the worker for the $(k+1)^{th}$ iteration.

Following [4], we denote the time taken to complete K iterations of local computation and wireless transmission, i.e., one instance of model training, as t, i.e., period. Note that t applies across all workers given that in the synchronous FL scheme, model training duration is constrained by the slowest worker, i.e., a worker has to wait for others to complete the training before the model can be aggregated. Moreover, each duration T can be represented in terms of instances of t (Fig. 1), and the time taken for worker i to collect and process the data for model training is denoted by a constant $c_i t$, $c_i > 0$. Without loss of generality, a model training request arrives at the beginning of each period. In the following, we consider the AoI and SL of the conventional FL scheme and the FL with caching scheme.

In the conventional FL scheme, the worker collects and processes the data on-demand upon the request of the model owner (Fig. 1a). As such, there is a constant minimum AoI regardless of the period in which the request arrives since data is collected only when it is required. The AoI is the time taken to complete FL training where

$$\bar{A}_i^c = t. \tag{1}$$

Moreover, the SL is given by the summation of periods taken for data collection and model training where

$$D_i^c = c_i t + t. \tag{2}$$

In contrast, with caching, the worker updates the cached data periodically every θ_i interval (Fig. 1b), independent of the period in which the request arrives, where

$$\theta_i = c_i t + a_i t, \; a_i \in \mathbb{N}, \tag{3}$$

and responds to the arriving request through local model training on the cached data. Note that we assume the ideal cache, i.e., each data owner is able to cache and update all the data, and leave the design of specific caching schemes for our future works. For each worker i, the T duration spans across $J_i = \frac{T}{\theta_i}$ update cycles.

Following the characteristics of the Poisson process, the probability of a request arrival is identical across periods and is given by $\frac{1}{T}$. If a request arrives at the n^{th} period during the data collection, the SL is $c_i t + t - (n-1)t$. Otherwise, if the request arrives at any of the remaining period within an update cycle, the SL is t. As such, the average SL $\bar{D}_s$ is

$$\begin{aligned} \bar{D}_s &= \frac{c_i}{c_i + a_i}\left(\cdots + (c_i t + t - (c_i - 1)t)\right)\frac{a_i}{c_i + a_i} t \\ &= \frac{c_i}{c_i + a_i}\left[\frac{c_i t}{2}(c_i + 3)\right] + \frac{a_i}{c_i + a_i} t. \end{aligned} \tag{4}$$

For content caching, the AoI of the data is at the minimum t only if the request comes during the data collection phase, or at the beginning of phase $(c_i + 1)t$. Otherwise, the AoI for a request that arrives at period lt will be $[l - (c_i + 1) + 1]t$, where $l \geq (c_i + 2)t$, i.e., the periods after data collection has been completed. As such, the average AoI is given by:

$$\begin{aligned} \bar{A}_s &= \frac{(c_i + 1)}{c_i + a_i} \cdot t + \sum_{l=c_i+2}^{a_i+c_i} \frac{1}{c_i + a_i} \cdot (l - c_i)t \\ &= \frac{t}{c_i + a_i}\left(c_i + 1 + \frac{(a_i - 1)(a_i + 2)}{2}\right). \end{aligned} \tag{5}$$

2

In comparison with the conventional FL, the FL with caching is a flexible system model that enables the model owner's management of the AoI and SL tradeoffs. From (4) and (5), we also observe the tradeoff between SL and AoI for our choice of cycle length θ_i. Intuitively, a lower θ_i, i.e., shorter cycle length or more update cycles J_i over T, enables lower average AoI given that data is more frequently updated. However, the SL increases as well since the updating takes time.

Given that the choice of θ_i involves a variation in resource expense of the workers, e.g, a lower θ_i represents higher data collection and caching cost incurred for worker i, an appropriate incentive mechanism design is required to motivate the workers towards a choice of θ_i that benefits the model owner. In the following, we model the profit functions of the workers and model owner respectively.

The data update cost η_i per update cycle is as follows:

$$\eta_i = \alpha_i \left(E_i^P + E_i^T + E_i^C \right), \tag{6}$$

where α_i refers to the unit cost of energy, E_i^P refers to energy consumed for traveling, E_i^T refers to energy consumed for wireless transmission of parameter updates to the model owner, and E_i^C represents computation energy[1]

With statistical information, the workers can be categorized into a set $\mathcal{N} = \{\eta_m : 1 \leq m \leq M\}$ of M data update cost types by data mining tools, e.g., k-means. The worker types η_m can be characterized by a probability mass function $p(\eta_m)$, where the cost types are indexed in a non-decreasing order $0 < \eta_1 \leq \cdots \eta_m \leq \cdots \leq \eta_M$. Therefore, the utility u_m of the worker type m is given as follows:

$$u_m(\omega_m) = R_m - \eta_m J_m, \tag{7}$$

where ω_m indicates the contract pair that consists of the rewards-update cycles bundle (R_m, J_m) designed for the type m worker, R_m refers to the contract rewards, and J_m refers to the number of update cycles.

To model the tradeoff between preferences for SL and AoI, the model owner profit function can be expressed by

$$\Pi = \sum_{m=1}^{M} Ip(\eta_m)\Bigg(\sigma\Bigg(w_a \Upsilon\left(1 + \frac{\mu}{\bar{A}_m(J_m)}\right) + w_d \Gamma\left(\frac{\phi}{\bar{D}_m(J_m)}\right)\Bigg) - R_m\Bigg), \tag{8}$$

where w_a and w_d represents the weighted preferences for information freshness, i.e., the inverse of AoI, and faster task completion, i.e., the inverse of SL, respectively. Both w_a and w_d are functions of J_m. Moreover, $w_a + w_d = 1$ and $w_d, w_a \in [0, 1]$. As an illustration, $w_a > w_d$ represents a model owner that values fresh information over faster task completion. In this regard, the model owner requires workers to have a higher J_m, i.e., more frequent data updating or a shorter cycle length θ_m equivalently. $\Upsilon(\cdot)$ is an increasing concave function with respect to the inverse of AoI to indicate the diminishing returns from information freshness, whereas $\Gamma(\cdot)$ is a linear function with respect to the inverse of SL. In addition, σ refers to profit conversion parameter from AoI and SL, μ and ϕ are system model parameters.

[1]Individually, each component is a function of other variables, e.g., the required propulsion power to balance the parasitic drag caused by skin friction and air of redirection [6]. We omit the details due to space constraints.

3 CONTRACT DESIGN

In this section, we discuss the conditions for contract feasibility. Then, we relax the constraints to derive the optimal contract $\Omega(\mathcal{N}) = \{\omega_m, 1 \leq m \leq M\}$.

3.1 Feasibility Conditions

A feasible contract must satisfy the following constraints:

Definition 3.1. Individual Rationality (IR): Each type m worker achieves non-negative utility if it chooses the contract item designed for its type, i.e., contract item ω_m.

$$u_m(\omega_m) \geq 0, 1 \leq m \leq M. \tag{9}$$

Definition 3.2. Incentive Compatibility (IC): Each type m worker achieves the maximum utility if it chooses the contract item designed for its type, and has no incentive to choose contracts designed for other types.

$$u_m(\omega_m) \geq u_m(\omega_{m'}), m \neq m',\ 1 \leq m \leq M. \tag{10}$$

The contract formulation is shown as follows:

$$\begin{aligned} &\max_{\Omega} \Pi(\Omega(\mathcal{N})) \\ &\text{s.t. } (9), (10). \end{aligned} \tag{11}$$

Lemma 3.3. *For any feasible contract $\Omega(\mathcal{N})$, we have $J_m < J_{m'}$ if and only if $R_m < R_{m'}$, $m \neq m'$.*

Proof. Due to space constraint, the proof is omitted. □

Lemma 3.4. *Monotonicity: For any feasible contract $\Omega(\mathcal{N})$, if $\eta_m > \eta_{m'}$, it follows that $J_m \leq J_{m'}$.*

Proof. Due to space constraint, the proof is omitted. □

From Lemma 3.3, we show the intuitive result that the IC contract offers higher rewards to workers which update the data more frequently, whereas Lemma 3.4 indicates that workers with lower cost of updating are willing to update the data more frequently. This gives us the necessary constraints.

Theorem 3.5. *A feasible contract must meet the following conditions:*

$$\begin{cases} J_1 \geq J_2 \geq \cdots \geq J_m \geq \cdots \geq J_M \\ R_1 \geq R_2 \geq \cdots \geq R_m \geq \cdots \geq R_M \end{cases} \tag{12}$$

Next, we further relax the IR and IC constraints. Intuitively, the minimum utility worker is the worker that incurs the highest cost of data update, i.e., the type M worker.

Lemma 3.6. If the IR constraint of the minimum utility worker, i.e., type M, is satisfied, the other IR constraints will also hold.

Proof. From the IC constraint and $\eta_m \geq \eta_M$, we have:

$$R_m - \eta_m J_m \geq R_M - \eta_m J_M \geq R_M - \eta_M J_M \geq 0.$$

As such, as long as the IR constraint of the type M worker is satisfied, the IR constraints of other workers will hold. □

Lemma 3.7. *(Reduce IC Constraints):* The IC constraints can be reduced into the Local Downward Incentive Constraints (LDIC).

Proof. Consider three worker types $\eta_{m-1} < \eta_m < \eta_{m+1}$. The two LDICs [5], i.e., constraints between type m and type $m-1$ workers , are provided as follows:

$$R_{m+1} - \eta_{m+1} J_{m+1} \geq R_m - \eta_{m+1} J_m, \text{ and}$$
$$R_m - \eta_m J_m \geq R_{m-1} - \eta_m J_{m-1}.$$

From Lemma 3.3, we have $R_m \geq R_{m+1}$ when $J_m \geq J_{m+1}$. As such, we can rewrite the LDICs as follows:

$$\eta_{m+1}(J_{m-1} - J_m) \geq \eta_m(J_{m-1} - J_m) \geq R_{m-1} - R_m \Rightarrow$$
$$R_{m+1} - \eta_{m+1} J_{m+1} \geq R_m - \eta_{m+1} J_m \geq R_{m-1} - \eta_{m+1} J_{m-1}.$$

As such, we have:

$$R_{m+1} - \eta_{m+1} J_{m+1} \geq R_{m-1} - \eta_{m+1} J_{m-1}.$$

Hence, if the LDIC constraint holds for type-m worker, it will also hold for type $m-1$ worker. This process can be extended downward from type $m-1$ to type 1 worker:

$$\begin{aligned} R_{m+1} - \eta_{m+1} J_{m+1} &\geq R_{m-1} - \eta_{m+1} J_{m-1} \\ &\geq \cdots \\ &\geq R_1 - \eta_{m+1} J_1. \end{aligned}$$

Given the monotonicity condition in Theorem 3.5, the LDIC also implies the LUIC and UIC holds. □

With Lemma 3.6, we reduce M IR constraints into a single constraint. With Lemma 3.7, we reduce $M(M-1)$ IC constraints into $M-1$ constraints. We are thus able to derive a tractable set of sufficient conditions for the feasible contract.

Theorem 3.8. A feasible contract must meet the following sufficient conditions:

(1) $R_1 - \eta_1 J_1 \geq 0$,
(2) $R_{m+1} - \eta_{m+1} J_{m+1} + \eta_m J_m \geq R_m \geq R_{m+1} - \eta_m J_{m+1} + \eta_m J_m$.

3.2 Contract Optimality

To solve the optimal contract rewards R_m^*, we first establish the dependence of optimal contract rewards $\mathbf{R}$ on the number of updates J. Then, we solve the problem in (11) with $\mathbf{J}$ only.

Theorem 3.9. For a known set of number of update cycles $\mathbf{J}$ satisfying $J_1 \geq \cdots \geq J_m \geq \cdots \geq J_M$ in a feasible contract, the optimal reward is given by:

$$R_m^* = \begin{cases} \eta_m J_m, & \text{if } m = M, \\ R_{m+1} - \eta_m J_{m+1} + \eta_m J_m, & \text{otherwise.} \end{cases} \tag{13}$$

Proof. We first assume there exists some $\mathbf{R}^\dagger$ that yields greater profit for the model owner, meaning that the theorem is incorrect, i.e., $\Pi(R^\dagger) > \Pi(R^*)$. This implies there exists at least a $t \in \{1, 2, \ldots, M\}$ that satisfies the inequality $R_t^\dagger < R_t^*$.
According to the LDIC constraint of Lemma 3.7,

$$R_t \geq R_{t-1} - \eta_t J_{t-1} + \eta_t J_t. \tag{14}$$

In contrast from Theorem 3,

$$R_t^* = R_{t+1} - \eta_t J_{t+1} + \eta_t J_t. \tag{15}$$

From (14) and (15), we can deduce that $R_{t+1}^\dagger < R_{t+1}^*$. Continuing the process up to $t = M$, we obtain $R_M^\dagger \leq R_M^* = \eta_m J_m$, which violates the IR constraint. As such, there does not exist $\mathbf{R}^\dagger$ that yields greater profit for the model owner. □

Following (13), we can re-express the optimal rewards as:

$$R_m^* = \eta_M J_M + \sum_{m=i}^{M} \Delta_t, \tag{16}$$

where $\Delta_M = 0$, $\Delta_t = -\eta_m J_{m+1} + \eta_m J_m$, and $t = 1, 2, \ldots, M-1$. By substituting the optimal rewards in (16) into the profit function of the model owner in (8) to derive $G_m(J_m)$, we obtain the following optimization problem:

$$\max_{(R_m^*, J_m^*)} \Pi(\Omega(\mathcal{N})) = \sum_{m=1}^{M} G_m(J_m),$$
$$\text{s.t. } J_1 \geq J_2 \geq \cdots \geq J_m \geq \cdots \geq J_M. \tag{17}$$

As such, J_m^* can be derived by separately optimizing each $G_m(J_m)$, e.g., through convex optimization tools, as follows:

$$J_m^* = \arg\max_{J_m} p(\eta_m)\left(\sigma\left(w_a \Upsilon\left(1 + \frac{\mu}{\bar{A}_m(J_m)}\right) + w_d \Gamma\left(\frac{\phi}{\bar{D}_m(J_m)}\right)\right) + \eta_{m-1} J_m \sum_{t=1}^{m-1} p(\eta_t) - \eta_m J_m \sum_{t=1}^{m} p(\eta_t)\right. \tag{18}$$

The derived solutions are feasible if and only if they satisfy the monotonicity constraint. Otherwise, we adopt the "Bunching and Ironing" algorithm for iterative adjustment.

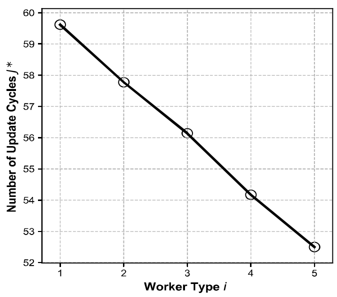


Figure 2: Update cycles vs. types.

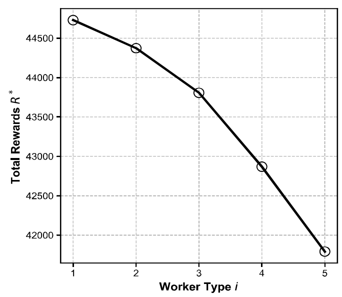


Figure 3: Rewards vs. types.

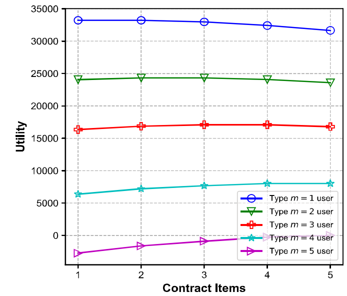


Figure 4: Utility vs. contract.

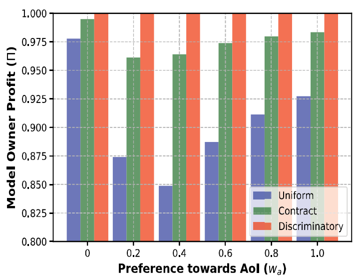


Figure 5: Profit comparison.

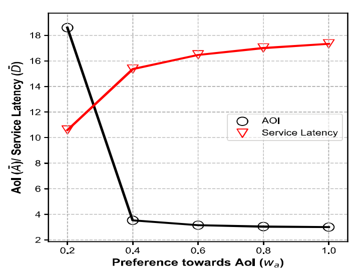


Figure 6: AoI and SL.

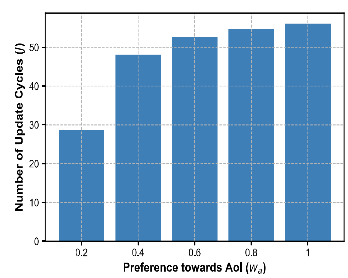


Figure 7: Number of update cycles.

Table 1: Table of Key Simulation Parameters

Simulation Parameters	Value
Update cost η	$\mathcal{N}(500, 300)$
Profit conversion parameters σ, μ	100, 000
Total time units T	1000
Periods taken for data aggregation c	5
Time unit per period t	3

3.3 Continuum Worker Types

We have only considered discrete worker types so far, i.e., workers with a fixed M types. In practice, there may be a continuum of worker types [5] with probability density function $f(\eta)$ and cumulative distribution function $F(\eta)$ bounded by $[\underline{\eta}, \bar{\eta}]$. The optimization problem can be rewritten as:

$$\max_{\{J(\eta),R(\eta)\}} \int_{\underline{\theta}}^{\bar{\theta}} [R(\eta) - \eta J(\eta)] f(\eta) d\eta,$$
$$\text{s.t. IR, IC constraints.} \tag{19}$$

Similarly, the IR constraint can be reduced to a single constraint involving the lowest utility worker, i.e., $u(\bar{\eta}) \geq 0$, whereas the IC constraints can be reduced as follows.

Lemma 3.10. The IC constraints can be reduced into the monotonicity, i.e., $\frac{dJ(\eta)}{d\eta} \geq 0$, and Local IC (LIC) constraints, i.e., $R'(\eta) = \eta J'(\eta) \frac{dJ(\eta)}{d\eta}$.

Proof. The monotonicity constraint can be derived following the procedures in Section 3.1. The LIC constraint [5] can be proven by contradiction, i.e., we assume there exists at least one η' which violates the IC constraint where

$$R(\eta) - \eta J(\eta) < R(\eta') - \eta J(\eta'),$$

which implies

$$\int_{\eta}^{\eta'} \left[R'(x) - \eta J'(x) \frac{dJ(x)}{dx} \right] dx > 0. \tag{20}$$

From the LIC, we have $\int_{\eta}^{\eta'} \left[R'(x) - xJ'(x) \frac{dJ(x)}{dx} \right] dx = 0$. For $\eta < x < \eta'$, the monotonicity condition implies $\eta J'(\eta) \frac{dJ(\eta)}{d\eta} \leq$

$xJ'(\eta)\frac{dJ(\eta)}{d\eta}$. As such, it follows that

$$\int_{\eta}^{\eta'} \left[R'(x) - \eta J'(x) \frac{dJ(x)}{dx} \right] dx < 0, \tag{21}$$

which contradicts with (20). Therefore, there does not exist a η' that violates the IC constraint. □

With the constraints relaxed, the optimization problem in (19) can be solved to derive the contract pairs $(J^*(\eta), R^*(\eta))$.

4 PERFORMANCE EVALUATION

To study the contract feasibility, we set $w_a = w_d = 0.5$. The simulation results in Fig. 2 and Fig. 3 validate the monotonic condition of the contract, which is consistent with Lemma 3.3. In addition, the higher rewards are distributed to worker types that incur lower marginal cost of data updating which is consistent with Lemma 3.4. The IC of our contract is also demonstrated in Fig. 4, where the utilities of each worker type are computed with the assumption that it takes on each of the contract items $1-5$. Clearly, each worker only derives the maximum utility when it takes on the contract item designed for its type.

We further compare the proposed incentive scheme with the uniform and discriminatory pricing scheme. In the uniform scheme, workers are offered the same contract bundle regardless of their types, i.e., the contract bundle for the minimum utility worker type. In the discriminatory scheme, we assume a hypothetical situation in which all worker types are known, i.e., information asymmetry does not exist. Then, we set the discriminatory scheme as the benchmark to compare the model owner profits among the three schemes in Fig. 5. Our proposed contract design allows a model owner to derive greater profits as compared to the uniform scheme, given that the self-revealing mechanism distinguishes between the worker types, thus validating that the adverse effects of information asymmetry is reduced.

In practice, a model owner may have different preferences for varying tasks. We vary the weights w_a and w_d within the range $[0.2, 1]$. In Fig. 6, the AoI and SL across varying preferences towards the AoI is plotted. Intuitively, a model owner that does not value information freshness but values that the request is met with lower SL will have a high AoI and low SL. Fig. 7 depicts the changes in the number of update cycles as the preference towards AoI varies. As expected, when the preference towards AoI is high, e.g., $w_a = 1$, the number of update cycles is the highest and the corresponding AoI is close to t.

5 CONCLUSIONS

In this paper, we propose the contract-theoretic incentive mechanism for FL in the sky considering the AoI and SL tradeoff for both the discrete and continuous workers. Performance evaluation has shown the IC and the flexibility towards varying AoI and SL requirements of our incentive scheme. For future works, we will consider the deviation from an ideal cache and design the efficient caching schemes.

6 ACKNOWLEDGEMENTS

This research/project is supported by the National Research Foundation (NRF), Singapore, under Singapore Energy Market Authority (EMA), Energy Resilience, NRF2017EWT-EP003-041, Singapore NRF2015-NRF-ISF001-2277, Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure NSoE DeST-SCI2019-0007, A*STAR-NTU-SUTD Joint Research Grant on Artificial Intelligence for the Future of Manufacturing RGANS1906, Wallenberg AI, Autonomous Systems and Software Program and NTU (WASP/NTU) under grant M4082187 (4080), and NTU-WeBank JRI (NWJ-2020-004), the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT20044), National Natural Science Foundation of China (Grant No. 51806157), and Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU JRI, NTU, Singapore.

REFERENCES

[1] Igor Kadota, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. 2016. Minimizing the age of information in broadcast wireless networks. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 844–851.

[2] Jakub Konečný, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).

[3] Wei Yang Bryan Lim, Jianqiang Huang, Zehui Xiong, Jiawen Kang, Dusit Niyato, Xian-Sheng Hua, Cyril Leung, and Chunyan Miao. 2020. Towards Federated Learning in UAV-Enabled Internet of Vehicles: A Multi-Dimensional Contract-Matching Approach. *arXiv preprint arXiv:2004.03877* (2020).

[4] Shan Zhang, Junjie Li, Hongbin Luo, Jie Gao, Lian Zhao, and Xuemin Sherman Shen. 2018. Towards fresh and low-latency content delivery in vehicular networks: An edge caching aspect. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 1–6.

[5] Yanru Zhang, Lingyang Song, Walid Saad, Zaher Dawy, and Zhu Han. 2015. Contract-based incentive mechanisms for device-to-device communications in cellular networks. *IEEE Journal on Selected Areas in Communications* 33, 10 (2015), 2144–2155.

[6] Zhenyu Zhou, Junhao Feng, Bo Gu, Bo Ai, Shahid Mumtaz, Jonathan Rodriguez, and Mohsen Guizani. 2018. When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning. *IEEE Transactions on Communications* 66, 11 (2018), 5526–5538.