Mobile Edge Computing, Blockchain and Reputation-based Crowdsourcing IoT Federated Learning: A Secure, Decentralized and Privacy-preserving System

Yang Zhao, *Student Member, IEEE*, Jun Zhao, *Member, IEEE*,
Linshan Jiang, *Student Member, IEEE*, Rui Tan, *Senior Member, IEEE*,
Dusit Niyato, *Fellow, IEEE*

***Abstract*—Internet-of-Things (IoT) companies strive to get feedback from users to improve their products and services. However, traditional surveys cannot reflect the actual conditions of customers' due to the limited questions. Besides, survey results are affected by various subjective factors. In contrast, the recorded usages of IoT devices reflect customers' behaviours more comprehensively and accurately. We design an intelligent system to help IoT device manufacturers to take advantage of customers' data and build a machine learning model to predict customers' requirements and possible consumption behaviours with federated learning (FL) technology. The FL consists of two stages: in the first stage, customers train the initial model using the phone and the edge computing server collaboratively. The mobile edge computing server's high computation power can assist customers' training locally. Customers first collect data from various IoT devices using phones, and then download and train the initial model with their data. During the training, customers first extract features using their mobiles, and then add the Laplacian noise to the extracted features based on differential privacy, a formal and popular notion to quantify privacy. After achieving the local model, customers sign on their models respectively and send them to the blockchain. We use the blockchain to replace the centralized aggregator which belongs to the third party in FL. In the second stage, miners calculate the averaged model using the collected models sent from customers. By the end of the crowdsourcing job, one of the miners, who is selected as the temporary leader, uploads the model to the blockchain. Besides, to attract more customers to participate in the crowdsourcing FL, we design an incentive mechanism, which awards participants with coins that can be used to purchase other services provided by the company.**

***Index Terms*—Blockchain, Crowdsourcing, Differential privacy, Federated learning, IoT, Mobile edge computing.**

I. Introduction

As the smart IoT device market goes competitive, IoT device manufacturers are seeking ways to break their bottleneck of current IoT device technology to provide better services to customers. The gap between manufacturers and customers' requirements is caused by the unfamiliarity of customers' habits of using IoT devices. To help IoT device manufacturers analyze and predict customers' usage of IoT devices, we design a crowdsourcing federated learning system.

Federated learning is a collaborative machine learning setting, and its goal is to train a global model by using distributed training belonging to clients with unreliable and relatively slow network connections [1], [2]. Federated learning assists in training a model using distributed data over a large number of consumers. Participants of the crowdsourcing federated learning task periodically exchange and update model parameters without sending their data. In many settings of federated learning, users download the neural network model, train the model locally, and upload their updates to the aggregator. The central aggregator will calculate the average of updates and apply it to the model. Hence, FL keeps the privacy and security of users' training data. Our system considers IoT devices of the same brand in a family as a unit, and one mobile is required to help collect data from IoT devices periodically and train the model. IoT manufacturers can extract useful data from customers' data for the detection, classification, and prediction of customers' next activities by using machine learning techniques [3]. The trained model will become more intelligent with more data generated by IoT devices. Since mobiles have limited computation power and battery life, it is challenging for many families to train the model locally. To encourage more customers to participate in the training task, we use the mobile edge computing server to enhance computation power for accelerating the training, which performs the training close to data sources. However, Melis *et al.* [4] demonstrate that gradient updates may leak information of participants' training data and develop passive and active inference attacks to exploit this leakage. Without security and privacy protection, many customers are unwilling to participate in federated learning. Besides, the collaborative approach to train the model is susceptible to model poisoning attack. Therefore, clients who participate in the federated learning and upload poisoning updates may affect the global model negatively [5]. Besides, potential security and privacy issues may also exist in the edge computing server, since the edge computing server belongs to a third party. The third party may sell or use customers' data in unauthorized ways, bringing unpredictable security issues to customers.

To solve the above privacy and security problems, we adopt the blockchain and differential privacy technologies when designing the system. The company uploads a preliminary

The authors are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

model with initialized parameters. The model is available on the blockchain for customers to download and train with their local data. The blockchain helps the crowdsourcing requester to audit whether there are malicious updates from participators. Traditional crowdsourcing system is hosted by the third party, which charges participants costly service fees, while our designed system uses blockchain to record crowdsourcing activities. Therefore, participants and the requester can save the high service fees while keeping the crowdsourcing system functioning. Due to the limitation of the block size, we propose to use the InterPlanetary File System (IPFS) [6] as the distributed storage solution when the model size is large. To protect the privacy of features, we adopt a partitioned deep model training approach [7] for customers. In the first step, customers extract features in the mobile from their data using the designed CNN network and add noise with formal privacy guarantee to perturb extracted features. In the second step, customers train dense layers of the model with perturbed features in mobile edge computing server. After training, customers should sign on hashes of encrypted models with their private keys and transmit locally trained models to the blockchain. Miners verify identities of senders, download models and calculate the average of all models' parameters to get the final model. One miner, selected as the temporary leader, encrypts and uploads the final model to the blockchain.

Besides, to motivate more customers to participate in the crowdsourcing task and reduce malicious and poisoning updates, we design a reputation-based crowdsourcing incentive mechanism. Customers will be rewarded with coins and reputation if they upload their models successfully. If a customer sends malicious data, his or her reputation will downgrade. The requester can choose reliable participants based on their reputation and previous performance. Even in the same task, participants with high reputation will get more rewards.

In this paper, we propose a mobile edge computing, blockchain, reputation-based crowdsourcing federated learning system. The system provides high computation power, confidentiality, audit, distributed storage and incentives for customers to participate in the collaborative federated learning, for smart IoT device manufacturers to better adjust their business plans by predicting customers' consumption behaviours.

Contributions. In this paper, we make the following contributions:

- A hierarchical collaborative learning system is proposed to help build the model for predicting customers' behaviours in operating IoT devices.
- Apply the differential privacy technique in training the federated learning model, so that an adversary cannot exploit the learned model to successfuly infer sensitive information of customers' data.
- Our blockchain-based system enhances the security of model updates by ensuring that malicious model updates are held accountable.

Organization. The rest of the paper is organized as follows. In Section II, we explain the technologies used in this paper. Section III presents related works and their deficiencies. We introduce our design of the system in Section IV. Section V shows the advantages and disadvantages of our designed system. Section VI presents experiments on real datasets. In Section VII, we conclude the paper and identify future directions.

II. Preliminaries, Assumptions, and Threat Model

A. Blockchain and InterPlanetary File System (IPFS)

The blockchain is a chain of blocks which contain the hash of the previous block, transaction information, and a timestamp. Blockchain origins from bitcoin network as an append-only, distributed and decentralized ledger to record peer-to-peer transactions permanently and immutably. The IPFS is a peer-to-peer distributed file system which enables distributed computing devices to connect with the same file system. We implement off-chain storage by using IPFS, and store hashes of data's locations on the blockchain instead of actual files. The hash can be used to locate the exact file across the system.

B. Differential Privacy

Differential privacy (DP) technique keeps the dataset's privacy while ensuring its utility. Intuitively, by incorporating some noise, the output of an algorithm under DP will not change significantly due to the presence or absence of one user's information in the dataset. By using the DP algorithm, analysts cannot derive an individual's information when analyzing the algorithm's outputs. Since its introduction, DP has received much interest in both academia and industry. Apple has incorporated DP into its mobile operating system iOS. Google has implemented a DP tool called RAPPOR in the Chrome browser to collect information about clients.

*Definition 1 (ϵ-**Differential Privacy (DP)** [8]):* A randomized algorithm Y to answer a query Q satisfies ϵ-differential privacy, if for any two neighboring datasets D and D' which differ in only one tuple, and for any possible subset of outputs $\mathcal{Y}$ of Y, the probability of $Y(D) \in \mathcal{Y}$ is at most e^{ϵ} times of the probability of $Y(D') \in \mathcal{Y}$, where ϵ is the privacy parameter.

Smaller privacy parameter ϵ means stronger privacy protection but less utility of Y as more randomness is introduced to Y. The ϵ is also often referred to as the privacy cost since smaller ϵ means stronger privacy protection, which can be understood as less privacy cost.

The **Laplace mechanism** of [8] can be used to ensure differential privacy by adding independent zero-mean Laplace noise with scale λ to each dimension of the output. Specifically, λ equals Δ/ϵ, where Δ is the ℓ_1-norm sensitivity of the query Q, which measures the maximum change of the true query output over neighboring datasets.

The **post-processing** property [8] of differential privacy means that a data analyst, without additional knowledge about the private database, cannot compute a function of the output of a differential private algorithm and reduce its privacy guarantee. Hence, in our design, although the noise is added to an intermediate layer of the neural network, the post-processing property ensures that the final trained model also satisfies differential privacy.

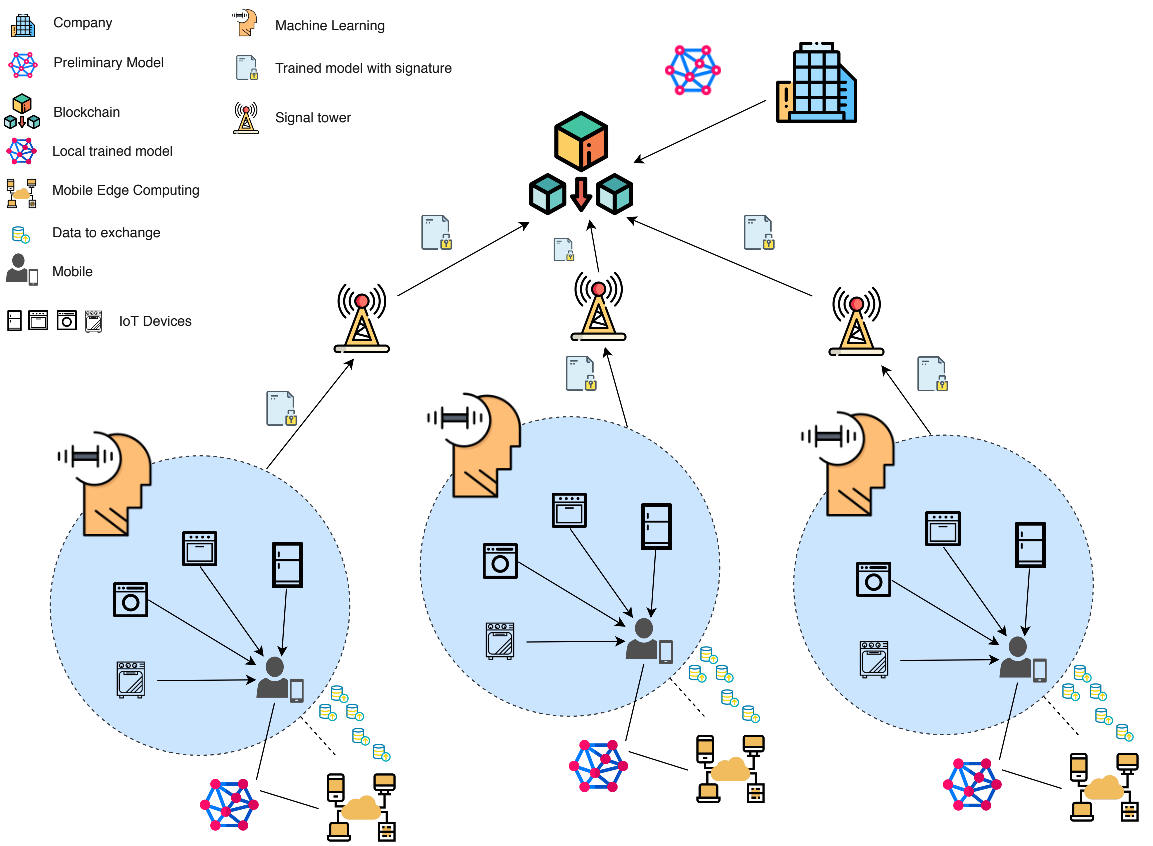

Fig. 1. An overview of our system, which we elaborate in Section IV on the next page.

C. System Assumptions

Our system makes the following assumptions. First, to use federated learning, we assume that customers are unwilling to share their raw data which may contain sensitive information. Second, the blockchain is permissioned so that only eligible customers can access. Third, participating customers are honest, and will not leak the company's model without approval.

D. Adversary Model

We consider the following adversary model. Some compromised or malicious customers may perform model poisoning attacks by sending malicious updates to the system. Our blockchain-based design makes these attacks accountable. The bad-behaving customers or external attackers may also launch information leakage attacks against a targeted peer victim. Incorporating differential privacy into the training process, together with the post-processing property of differential privacy, ensures the privacy of the training dataset.

III. Related Work

Weng *et al.* [9] proposed a deepchain system for collaborative learning. They also designed an incentive mechanism for attracting parties to participate in learning. However, they did not consider the malicious attack during transmitting gradients and learned model. We use the signature for receivers to verify the identity of senders to preclude impersonating attackers who send data to the blockchain. Jiang *et al.* [7] partitioned the deep neural network model in training to protect features' privacy. From [7], the feature extraction is performed locally in the edge devices while the classification is executed in the cloud service. To protect the privacy contained in the features sent from the edge devices against the curious-but-honest cloud, the approach perturbs the features before the transmission. In this paper, we propose a hierarchical collaborative learning system, in which the partitioned deep model training is performed between the end devices and the edge computing server whereas the models built by the edge servers are sent to the blockchain to perform federated learning. As a new contribution of this paper, the federated learning in the blockchain well addresses the training data imbalance issue. Thus, the partitioned deep model training approach in [7] is used as a building block of our system. In particular, we improve the normalization technique proposed in [7] that leads to substantial learning accuracy improvement. The details of the improvement will be presented in Section IV-C. Wang *et al.* [10] designed the *In-Edge AI* framework to use the distributed end devices and edge nodes to train a model while reducing the communication cost during exchanging parameters. Nonetheless, they did not address any security and privacy issues during model aggregation. Li *et al.* [11] designed a blockchain-based decentralized framework for crowdsourcing, which enables crowdsourcing without the centralized server. They do not use differential privacy and signature techniques as in our paper. Konečnỳ *et al.* [1] and McMahan *et al.* [2] proposed the concept of federated learning, but their work did not consider privacy or security during federated learning.

IV. System Design

We design a system for the smart IoT device manufacturers who are interested in building a predictive model using data from clients' IoT devices to predict the trend of customers' consumption. The system consists of three main components: the requester, participants, and a blockchain, reputation-based crowdsourcing system. Figure 1 on the previous page presents an overview of our system.

A. Requester - Company

Consider building the predictive model as a crowdsourcing federated learning task. The company serves as the requester, and customers act as participants. When the company raises a crowdsourcing request for federating learning, customers who have the required devices can apply to participate in the crowdsourcing task. The crowdsourcing system implemented with the blockchain will record the progress of the crowdsourcing task. As the requester, the company should design an incentive mechanism and upload an initial model to the blockchain.

1) Incentive mechanism: An incentive mechanism is necessary for collecting more data. Since customers' data contain personal privacy, many people are unwilling to contribute data to building the model. With an incentive mechanism, participants can get rewards or penalties based on their contributions. After purchasing the home appliances, customers may concern the quality and warranty of those devices. For manufacturers, they have enough resources to provide after-sales service. Therefore, after customers contribute to crowdsourcing federated learning task, they will be rewarded with coins. The company distributes these coins so that customers can use coins to trade for services provided by manufacturers including maintenance and upgrading of devices. Besides, we also design a reputation-based crowdsourcing mechanism. In the beginning, every customer's family will have an initial reputation value. When customers contribute correct and useful model parameters, their reputation will increase. In contrast, if they are caught to upload malicious models, their reputation will get deducted. Customers with higher reputation will have more opportunities to participate in the next crowdsourcing task and gain more rewards. The reputation status is recorded by the blockchain.

2) Company puts preliminary model in the blockchain: To facilitate the progress of federated training, we use the blockchain to store the initial model. Otherwise, the company needs to send the model to everyone or save it in a third party's cloud storage. The initial model's parameters are randomly selected. Since many customers participate in the crowdsourcing federated learning, the company can learn a satisfactory model.

B. Crowdsourcing system host - Blockchain

By using the blockchain, the crowdsourcing system can store models permanently. A company can use blockchain to save both the initial model and the final model. Customers can upload their locally trained models to the blockchain. Because of the limitation of the block size, we propose to use IPFS as the offline storage solution. We can store the actual data locally. But the hash of the file location is saved in the blockchain. When accessing the blockchain, customers can get the hash of the file location. Then they can use the hash to get the actual file peer-to-peer. Miners are responsible for confirming transactions and calculating the averaged model parameters. If participants upload their local trained models to the blockchain, miners verify that uploaded models' signatures are valid and confirm the transaction. After all the participants upload their trained models, workers download them and start calculating the averaged model parameters. Then one of the workers is selected as the leader to upload the final model to the blockchain.

1) Miners verify the uploaded file: When a participant tries to upload his model to the blockchain, a miner checks the signature of the uploaded file. If the signature is valid, the miner confirms that the file is from the legal participant and puts the transaction in the transaction pool. Then one of the miners will generate the block and approve the transaction. If the signature is invalid, the miner should reject the transaction, because it is possible that an adversary use the public key encrypting the faked model and upload it to the blockchain to attack the federated learning model.

2) A selected miner updates the model: Miners are also responsible for calculating the average of uploaded models' parameters to get the FL model. We assume there is a program for every miner to get correct averaged parameters. Miners compete for updating parameters to get the reward. Motivated by Algorand [12], we use Verifiable Random Functions (VRF) as a local and non-interactive way to select a subset of users as the leader candidates (weighed by their coins) and determine their priorities. A leader candidate with the highest priority will become the leader to update the model parameters. As each user is weighted by their coins, one unit of coin can be regarded as a sub-user. A user with w coins has w "sub-users". Let τ be the expected number of sub-users that the system desires to choose, and W be the total mount of coins of all users. Then the probability p of any coin being chosen can be set as τ/W. For a user u with w units of currency, it will first use its secret key to generate a hash and proof via VRF. The interval $[0,1]$ is divided into $w+1$ sub-intervals, so that the number j of selected sub-users for this user is determined by which sub-interval $hash/2^{hashlen}$ falls in ($hashlen$ denotes the length of $hash$); i.e., j satisfies $hash/2^{hashlen} \in [\sum_{k=0}^{j} \binom{w}{k} p^k (1-p)^{w-k}, \sum_{k=0}^{j+1} \binom{w}{k} p^k (1-p)^{w-k})$ (if $hash/2^{hashlen}$ equals 1, then j equals w). Other users can use the proof to check that user u indeed has j sub-users selected. The number of selected sub-users is each user's priority. The user with the highest priority will become the leader. The selected leader is responsible for updating the model parameters, and uploads the final model to the blockchain.

C. Participants - Customers

Customers who have home appliances satisfying crowdsourcing requirements can apply for participating in the crowdsourcing task. Based on the incentive mechanism, participants can get coins and the respective reputation based on their contributions. Large home appliance manufacturers always produce a variety of IoT devices. Different IoT devices contain different storage and computation power. Therefore, it is difficult to enable every IoT device to train the deep model. To address this, we adopt the partitioned deep model training approach proposed in [7]. Specifically, we use a mobile to collect all IoT devices' data and extract features on the mobile first. To preserve privacy, we add ϵ-DP noise to the features. Then customers can continue to train dense layers in the mobile edge computing server. We clarify participants' responsibilities in four detailed steps as following:

1) Customers download the initial model from the blockchain: Participants of the crowdsourcing federated learning should check and download the initial model which is available in the blockchain. Then they can start preparing for training data.

2) Customers extract features on the mobile: The mobile phone collects all participating IoT devices' data periodically. After collecting an amount of data, the customer can start training the model. Since the mobile edge computing server is provided by a third-party, there is a potential security issue if the feature extraction based on the raw data is performed on the edge computing server. Therefore, the local training system consists of two phrases: the mobile training and the mobile edge computing server training. Because perturbing original data directly may compromise the accuracy and performance of federated learning model, we use the original data in the CNN layers to extract features in the mobile. Then, we add ϵ-DP noise to the features before they are used to train the dense layers.

3) Customers train dense layers in the mobile edge computing server: The mobile sends privacy-preserving features and original labels to the mobile edge computing server so that the edge computing server can help train the fully-connected layers and compute the training loss. The perturbed features serve as the input of dense layers. The training loss is finally fed back to the mobile to update the front layers.

4) Customers upload models to the blockchain: After training the model, participants should sign on the hash of the model with their private keys, and then upload models to the blockchain. However, if miners find out the signature is invalid, the transaction fails, because it is possible that an adversary tries to attack the learning process using faked data. After miners confirm the transaction, customers can use the transaction history as an invoice to claim reward and reputation. By using the immutable property of the blockchain, both manufacturers and participants cannot deny the transaction. The company can download and check customers' updates to verify whether there are malicious updates. Malicious customers' reputation should be deducted if they are caught by the company. Customers with low reputation will have a lower chance to participate in the crowdsourcing job in the future.

We now present the improvement that we make to the normalization technique proposed in [7]. Although the CNN can have many channels, our analysis below focuses on one channel only for simplicity. For this channel, suppose the output of the convolutional layers has dimension $L \times W$. Let the value at a position $\langle i, j \rangle$ for the feature of image k be $X_{i,j,k}$. Given i and j, Jiang *et al.* [7] adopt batch normalizaiton, which transforms $X_{i,j,k}$ to $\widetilde{X}_{i,j,k}$ so that for each batch B, the values $\widetilde{X}_{i,j,k}$ for $k \in B$ have a zero mean and a variance of 1; i.e., $\frac{1}{|B|} \sum_{k \in B} \widetilde{X}_{i,j,k}$ equals 0 while $\frac{1}{|B|} \sum_{k \in B} (\widetilde{X}_{i,j,k})^2$ equals 1. From the Cauchy–Schwarz inequality, [7] bounds $\widetilde{X}_{i,j,k}$ in the interval $[-\sqrt{N-1}, \sqrt{N-1}]$ for any i, j, k so that if one value in the feature $\big\{X_{i,j,k} \mid i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}\big\}$ of image k varies, the sensitivity of $\big\{\widetilde{X}_{i,j,k} \mid i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}\big\}$ is no greater than $2\sqrt{N-1}$. Then, according to the Laplace mechanism [8], independent zero-mean Laplace noise with scale $2\sqrt{N-1}/\epsilon$ is added to each $\widetilde{X}_{i,j,k}$ for $i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}$, to protect $X_{i,j,k}$ under ϵ-differential privacy. In our approach, we normalize $X_{i,j,k}$ for $i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}$ as $\hat{X}_{i,j,k}$ in the interval $[-\sqrt{N-1}, \sqrt{N-1}]$ so that if one value in the feature $\big\{X_{i,j,k} \mid i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}\big\}$ of image k varies, the sensitivity of $\big\{\hat{X}_{i,j,k} \mid i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}\big\}$ is $2\sqrt{N-1}$. Then according to the Laplace mechanism [8], independent zero-mean Laplace noise with scale $2\sqrt{N-1}/\epsilon$ is added to each $\hat{X}_{i,j,k}$ for $i \in \{1, 2, \ldots, L\}$ and $j \in \{1, 2, \ldots, W\}$, to protect $X_{i,j,k}$ under ϵ-differential privacy. From the above discussions, batch normalizaiton of [7] enforces not only $\widetilde{X}_{i,j,k} \in [-\sqrt{N-1}, \sqrt{N-1}]$ but also the mean $\frac{1}{|B|} \sum_{k \in B} \widetilde{X}_{i,j,k}$ to be 0 and the variance $\frac{1}{|B|} \sum_{k \in B} (\widetilde{X}_{i,j,k})^2$ to be 1, while our normalizaiton technique requires only $\hat{X}_{i,j,k} \in [-\sqrt{N-1}, \sqrt{N-1}]$ without any contraints on the mean and variance. Experiments to be presented in Section VI show that our normalizaiton technique significantly improves the learning accuracy over that of [7].

V. Pros and Cons of our framework

A. Privacy and Security

Our system employs the differential privacy technique to protect the confidentiality of features and the security of the model respectively. Thus, to some extent, the system keeps the participating customers' data confidential. Besides, during the training, we enhance the protection of privacy by adding Laplacian noises to the extracted features. Furthermore, the trained model is encrypted and signed by the sender to prevent the attackers and imposters from stealing the model or derive original data through reverse-engineering.

B. Delay in Crowdsourcing

Assume that there is a large sum of participants, and the system highly depends on participants' training results to get the predictive model in the one round. Unlike other crowdsourcing jobs, the company in our system prefers participants to follow their lifestyle instead of rushing to finish the job to get the real status. As a result, participants who seldom use devices may postpone the overall crowdsourcing's progress.

TABLE I
THE CNN STRUCTURE USED IN OUR EXPERIMENTS. THE NORMALIZATION LAYER AND THE NOISEFICATION FOLLOWED ARE USED TO ACHIEVE THE DIFFERENTIAL PRIVACY (DP) GUARANTEE.

30 3×3 conv filters	pooling (2×2)	80 3×3 conv filters	pooling (2×2)	Normalization layer	noisefication for DP	⇒	3920 ReLUs	600 ReLUs	100 ReLUs	30 ReLUs	20 ReLUs	10 ReLUs	softmax
Conv Layers				For DP			Dense layers						

VI. EXPERIMENTS

We conduct experiments about our designed federated learning with the differential privacy method using the MNIST handwritten image dataset [13].

A. Experiment Setup

We use the public dataset MNIST which includes 60,000 training image samples and 10,000 test image samples for the experimental evaluation. Each sample is a 28×28 gray scale image showing a handwritten number within 0 to 9. Our designed CNN network includes hidden layers which are responsible for feature extraction and dense layers for classification. We have two convolutional layers with 30 and 80 channels respectively. After each convolutional layer, we deploy a max-pooling layer to reduce spatial dimensions of the convolutional layers' output. Therefore, max-pooling layers help to accelerate the learning speed of the neural network. Normalization is used after all non-linear layers, i.e., convolutional layers. Normalization layer enables the computation of sensitivity in differential privacy to determine the amount of noise addition, speeds up the learning rate, and regularizes gradients from distraction to outliers. Then, we apply ϵ-DP noise to perturb the output of normalization layers to preserve the privacy of features. Perturbed features serve as input of dense layers for classification in the mobile edge computing server. In our designed model, the dense layers contain four hidden layers. The dimension decreases from 3920 to the dimension of the label which is 10. Finally, there is a softmax layer to predict label and compute loss. The architecture of CNN is shown in Table I. We simulate the federated learning by constructing the model using the averaged parameters of multiple trained model parameters.

In our experiment, we set the hyperparameters of CNN as follows: the learning rate is 0.01, and the batch size N is 64. We set the range of ϵ to [1, 10]. Before training, we separate the training image dataset into equally ten parts, meaning that each participant gets 6000 training images randomly. We normalize each dimension of the feature to the interval $[-\sqrt{N-1}, \sqrt{N-1}]$ for N denoting the batch size, so that the sensitivity of the normalized feature vector when one dimension of the feature changes is $2\sqrt{N-1}$. Then according to the Laplace mechanism [8], independent zero-mean Laplace noise with scale $2\sqrt{N-1}/\epsilon$ is added to each dimension of the normalized feature, to protect the feature under ϵ-differential privacy.

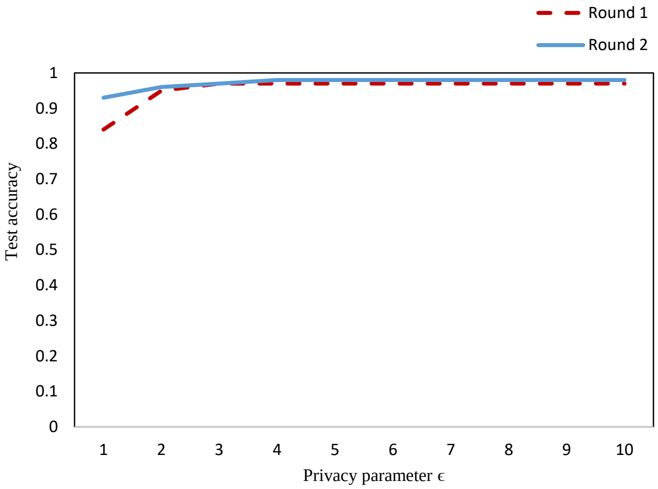

Fig. 2. Impact of differential privacy (DP) parameter ϵ on the test accuracies in Round 1 and Round 2 of the federated learning model.

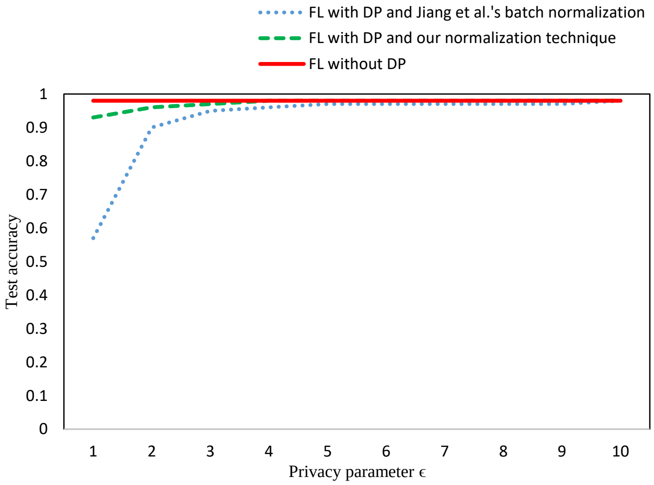

Fig. 3. Comparing the test accuracies between federated learning (FL) without differential privacy (DP) and different DP-aware FL algorithms, including DP-aware FL using our normalization technique, and DP-aware FL using Jiang *et al.* [7]'s batch normalization.

B. Experimental Results

Figure 2 shows how the privacy parameter ϵ affects the test accuracy of federated learning. In our experiment, we run the federated learning process for two rounds to verify that our designed approach is practical. The test accuracy increases with the privacy parameter ϵ. Larger ϵ means that less noise is added to features so that the privacy protection is weaker. In the literature, typical ϵ values chosen for experimental studies are between 0.1 and 10 [14]. Our experiment shows that we can achieve at least 90% accuracy in two rounds when the privacy parameter ϵ is at least 1. Before training, we initialize a model with random parameters, and the model with initial parameters will be used by all parties for their local training. After the first round, we get a new model by averaging all parties' models from the first round. Then in the

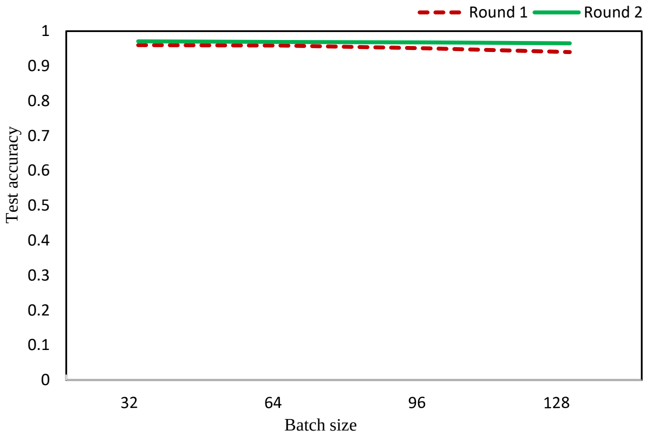

Fig. 4. Impact of batch size on the test accuracy of the federated learned model with ϵ-differential privacy for ϵ being 2.

second round, parties start training using the averaged model from the first round. Through our experiment, we can verify that our designed federated learning method is effective.

Figure 3 compares the test accuracies between federated learning (FL) without differential privacy (DP) and different DP-aware FL algorithms, including DP-aware FL using our normalization technique, and DP-aware FL using Jiang *et al.* [7]'s batch normalization. In all experiments of Figure 3, we report the accuracy by running the FL for two rounds. Figure 3 shows the superosity of DP-aware FL using our normalization technique over DP-aware FL using [7]'s batch normalization. For each DP-aware FL, we also observe that the test accuracy gets closer to the test accuracy of FL without DP as the privacy parameter ϵ increases.

Figure 4 presents the effect of batch size on the test accuracy. We test on four different batch sizes namely 32, 64, 96 and 128. The privacy parameter ϵ is set as 2. As shown in Figure 4, a smaller batch size induces a higher test accuracy in the first round. The explanation is as follows. Recall that independent zero-mean Laplace noise with scale $2\sqrt{N-1}\big/\epsilon$ is added to each dimension of the normalized feature, to protect the feature under ϵ-differential privacy. Hence, a smaller batch size means less noise added to the normalized feature and hence better accuracy, as illustrated in the plots for the first round in Figure 4. From Figure 4, we also observe that the batch size does not impact the accuracy much in the second round.

VII. Conclusion and Future Work

In this paper, we present a reputation-based crowdsourcing federated learning system for IoT manufacturers to provide better servicess in the future. We use multiple state-of-the-art technologies to construct the system, including mobile edge computing, blockchain, distributed storage, and federated learning. Besides, our system enforces differential privacy to protect the privacy of customers' data. By designing a proper incentive mechanism for the crowdsourcing task, customers are willing to participate in the crowdsourcing tasks. The blockchain will audit all customers' updates during the collaborative training so that the system can hold malicious updates accountable. Future directions include conducting more experiments and testing our system with real-world IoT device manufacturers.

References

[1] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[3] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE Conference on Computer Communications (INFOCOM)*, 2018, pp. 63–71.

[4] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[5] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[6] J. Benet, "IPFS-content addressed, versioned, P2P file system," *arXiv preprint arXiv:1407.3561*, 2014.

[7] L. Jiang, X. Lou, R. Tan, and J. Zhao, "Differentially private collaborative learning for the IoT edge," in *International Workshop on Crowd Intelligence for Smart Cities: Technology and Applications (CISC)*, 2018.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, 2006, pp. 265–284.

[9] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *Cryptology ePrint Archive, Report 2018/679*, 2018.

[10] 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 Network*, 2019 (to appear).

[11] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. Deng, "CrowdBC: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[12] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *ACM Symposium on Operating Systems Principles (SOSP)*, 2017, pp. 51–68.

[13] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," 2010, accessed on March 1, 2019. [Online]. Available: http://yann.lecun.com/exdb/mnist

[14] D. Sánchez, J. Domingo-Ferrer, and S. Martínez, "Improving the utility of differential privacy via univariate microaggregation," in *International Conference on Privacy in Statistical Databases*, 2014, pp. 130–142.