A Light-Weight Crowdsourcing Aggregation in Privacy-Preserving Federated Learning System

Ke Zhang, Siu Ming Yiu
The University of Hong Kong
Hong Kong, China
{kzhang2,smyiu}@cs.hku.hk

Lucas Chi Kwong Hui
Hong Kong Applied Science and Technology Research Institute
Hong Kong, China
lucashui@astri.org

***Abstract*—Federated Machine Learning (FML) sheds light on secure distributed machine learning. However, generic FML methods may lead to privacy-leakage through the sharing of training information of individual models and have relatively poor performance when the training datasets for individual models are biased and diversified. This is a problem in combining models trained in different scenarios of IoT devices since the available training datasets are usually limited and biased. To tackle this problem, we propose a novel approach to precisely ensemble results from different models in distributed edge devices. Instead of passing the training information of individual models around that requires a relatively large amount of bandwidth and compromises data privacy, we suggest employing a trusted central agent that only collects different inference results from edge devices. Then based on a limited amount of labeled data, the agent runs a designed statistical iterative crowdsourcing algorithm to combine results for a more accurate aggregated prediction towards a user query. Our proposed system model, "Privacy-Preserving Federated Learning System", together with our light-weight Secure Crowdsourcing Aggregation (SC-Agg) algorithm, provide a more accurate prediction for outside queries at little cost without any prior knowledge of what query will be submitted. We experimentally verify that in our system, SC-Agg consistently outperforms the majority voting method and the best performing model of the ensemble in all testing scenarios. We believe that SC-Agg fits the real-world IoT applications better than other methods, such as the vanilla majority voting, for its robustness and better performance.**

***Index Terms*—Federated Learning, Crowdsourcing, Privacy, IoT**

I. Introduction

An emerging trend nowadays is to embed machine learning models into IoT (Internet of Things) devices, such as drones, to promote their capabilities in daily application. However, these models usually do not perform well because of both the hardware constraints and limited training data. In terms of available training datasets, IoT devices of a particular data owner can only collect data specific to its scenarios. This makes the general deployment a difficult task, as the inferences of these models are usually biased to the distribution of the available training datasets. Improving the overall inference performance of these models in edge devices under limited and biased training data becomes an essential problem to be addressed immediately.

One natural idea is to improve the training step by using all data records collected from multiple data owners. Because of the privacy issue and profit conflicts, data owners are reluctant to share their data, especially when the data contains sensitive information of their clients. Thus, the approach of Federated Machine Learning (FML) was proposed [1]–[3]. In a typical FML system, each entity shares training information, such as gradient values [4], with one another so that each can update their models using the shared information from other models. Intuitively, the final "assembled" model of each entity could be similar to an ideal model trained by all collected data from all owners. FML brings synergy for big organizations in different realms in the sense that they do not need to share sensitive data. At the same time, the resulting model can perform better and may perform similarly as an ideal model trained by multiple datasets provided by different owners [5]. Applying FML on edge devices, however, exhibits a few issues to be tackled. Biased and distributed training datasets greatly degenerate the overall performance of FML [6], while this diversity is widespread for IoT devices, especially for those customized products. FML requires a relatively large amount of communication bandwidth for edge devices to formulate the final model. Sharing training information draws attention to attackers too. Recently, several studies [4], [7] have proposed attacks based on the shared model parameters in FML methods. It is desirable to come up with different approaches that still can make use of FML ideas to tackle these problems (e.g., [6], [8], [9]).

In this paper, we propose a novel idea to solve the above problem. Instead of sharing machine learning model's parameters, we utilize a trusted central agent (or server) to collect only the inference results from individual models. We denote our system as "Privacy-Preserving Federated Learning System". In our system, the agent between users and edge devices blocks outsiders' direct accesses to distributed models. All edge devices do not need to communicate with one another to update or construct models. Outside users will only communicate with the agent. For every query from a user, edge devices only need to provide their inference results to the agent. The agent owns a limited set of labeled data (either publicly available data or received from distributed data owners) for precise aggregation. Our approach can reduce the transmitted information from the edge devices and also avoid attacks on sharing individual models' parameters or related information. Under this framework, we also propose a

"Light-Weight Secure Crowdsourcing Aggregation" (SC-Agg) algorithm to evaluate and aggregate distributed information collected from edge devices. To elevate the robustness of FML encountering models with biased inference capacities, we utilize the crowdsourcing concept with modifications that fit into generic FML. Technically speaking, we enhance the fundamental crowdsourcing approach by aggregating responses based on a capacities-reflecting weights matrix. The weights matrix is statistical iteratively calculated based on remote models' responses and the queried data. Existing crowdsourcing methods that also involve evaluations for labelers have a strong assumption of possessing prior knowledge for queries. However, SC-Agg requires no historical information about past queries. Rather than occupying a large amount of data transmission bandwidth, computing resources and hardware storage to train a new model or update distributed models as in the FML, communication complexity of SC-Agg is as low as $\mathcal{O}(N)$, where N is the number of involved distributed weak models. Without additional training operations or massive communications, SC-Agg is a light-weight algorithm for both the central server and end devices.

We experimentally verify that for queries in different distributions, SC-Agg consistently outperforms the majority voting and best ensemble model. Comparing to FML, SC-Agg does not need to create any final model using multiple models created by different end devices, thus minimizing the communication overheads of the devices while trying to eliminate the biased performance of different models due to limited training datasets. We believe that SC-Agg is robust and can be applied to different real-world application scenarios.

We summarize our main contributions of this paper as follows:

- We construct a Privacy-Preserving Federated Learning System, ensuring the confidentiality of both distributed models and corresponding sensitive training data.
- Based on the Privacy-Preserving Federated Learning System, we propose a light-weight Secure Crowdsourcing Aggregation (SC-Agg) algorithm to solve the aggregation problem for distributed weak models with diversified inference capacities.
- We experimentally verify that SC-Agg outperforms both the best model among the ensemble and the majority voting approach for all our test scenarios.

II. Related Works

A. Federated Machine Learning

Federated Machine Learning (FML) is categorized into three types, *i.e.*, Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL). HFL fits the scenarios that distributed models are trained with datasets sharing the same feature space but different samples [3]. VFL is for distributed data owners who have overlapping data records but different attributes for each record. FTL applies to the scenarios that the two data sets differ not only in samples but also in feature space.

In a distributed system such as IoT, however, remote IoT devices for one specific application have little training data in common. For those IoT devices embedded with models aiming at the same inference task, their feature spaces are usually the same. The problem we are going to solve in the distributed IoT system is similar to the HFL application scenario. In typical HFL, distributed model owners send encrypted sensitive training data or the encrypted gradient descent to the center server for updating their models' parameters [5]. Inspired by HFL, we design the "Privacy-Preserving Federated Learning System" by introducing a central server into a distributed IoT system.

General HFL methods using homomorphic encryption for communication takes tremendous computation resources, which is impractical for edge devices. The distributed models need to be trained with the same learning algorithm and constructed in the same structure to ensure the correctness of the HFL process. These factors and constraints are not only overwhelmed for IoT end-devices for their limited computation abilities but also tricky for generalization in real-world practice. Primarily, HFL relies heavily on the models' internal information retrieved from distributed models. Because the central server utilizes the responses from data owners without evaluating, it accelerates the probability for the joint global model being inserted with backdoors, and it is possible for adversaries carrying out data poisoning attack [10].

The SC-Agg algorithm we propose for our system aims at aggregating can be operated on frozen distributed weak models ensemble. We only query for the inference results from models in a black-box manner. In SC-Agg, by not sharing models private information, we not only protect the privacy of distributed models but also lower the probability of being attacked [10].

B. Crowdsourcing

Our research differs from FML in the sense that, in order to minimize the degeneration in FML brought by the diversified inference capacities of distributed weak models, we further exploit the crowdsourcing concept to aggregate inference results more precisely through carrying out evaluations for involved models and the queried data.

Crowdsourcing is a branch of ensemble learning, where models for different classification tasks contribute efforts to generating comprehensive predictions. Most state-of-art crowdsourcing learning methods reach better results by analyzing both models' capabilities and the difficulty level of labeling queried data. Some require a large amount of labeled data for precise models' capacities estimation [11], [12], which is impractical and expensive in real-world applications. Others achieve accurate predictions by referring to historic queries [13] as prior knowledge of coming queried data. In real-world applications, unfortunately, responses from the public can be noisy and incorrect, while useful reflections are hard to identify and retrieve.

Due to both the lack of reliable feedback from outside and the difficulty in gaining a large amount of labeled data in

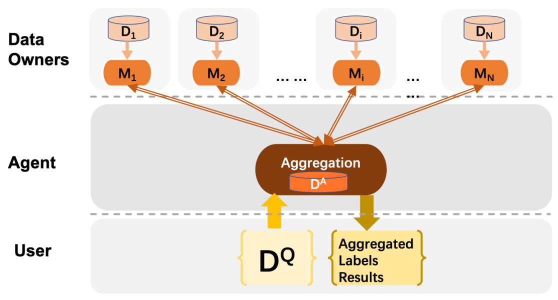

Fig. 1. Privacy-preserving federated learning system

real-word IoT application situations, we design an SC-Agg algorithm for our "Privacy-Preserving Federated Learning System" with further adjustments based on existing crowdsourcing methods. We discard exploiting historical queried data, and instead, we utilize only a small labeled dataset for accurate models' capacities evaluation and inference aggregation. These two improvements lower the requirements for well-performed crowdsourcing while elevating the result's quality of generic FML in real-world IoT application scenarios.

III. Privacy-Preserving Federated Learning System

In this section, we describe the construction of our Privacy-Preserving Federated Learning System in detail. There are three roles in the system, the **DO** (Data Owner), the **User**, and the **Agent**. Similar to a generic FML, our system introduces a trusted central server, *i.e.*, **Agent**. **Agent** blocks outside **User**s from directly accessing distributed weak models, *i.e.*, **DO**s. **Agent** is responsible for aggregating all the noisy responses from distributed **DO**s and computing the final predictions for the **User**'s query with SC-Agg. Fig. 1 shows the overview.

- **DO** (Data Owner): **DO**s are the distributed data owners in our system. There are several **DO**s in our system. Every **DO** privately trains a weak machine learning model with its sensitive data. **DO**s do not share training data or the models directly with others for privacy concerns. Note that **DO**s decide their training algorithms, and the distributions of their training dataset are diversified and unpredictable. Consequently, the performance of each weak classifier varies in different inference tasks.
- **User**: There can be several public **User**s sending queries. Our system processes one **User**'s query at a time. **User** queries **Agent** with an unlabeled dataset and obtains corresponding inference results from **Agent**. Note that **User** does not directly communicate with **DO**s, while **Agent** is the only interface for **User** to access. Moreover, the distributions of **User** query data are random, *i.e.*, neither **Agent** nor **DO**s can predict the data distributions of queries based on previous experience. Also, **User** will not provide responses for **Agent** to refine **Agent**'s future predictions.
- **Agent**: There is only one **Agent** in our system. **Agent** is an intermediary between **DO**s and **User**. Introducing **Agent** into our system benefits both efficiency and privacy. **Agent** is responsible for aggregating predictions from **DO**s into final results, which eliminates communications among distributed **DO**s. Also, **Agent** prevents **User** from querying **DO**s directly to analyze sensitive training data [14]. **Agent** owns a small public labeled dataset either collected publicly or provided by **DO**s. Note that this dataset contains no sensitive data, and its distribution cannot reveal other datasets' distributions in our system. This small labeled dataset is utilized for estimating both the distribution of queried data, and remote models' capacities for queried data. The detail process is described in Section V-A.

There are only two types of communications in our system, 1) ***Agent** with **DO**s:* After **Agent** sending unlabeled data to every **DO**s, **DO**s send back corresponding inference probabilities distributions to **Agent**; and 2) ***Agent** with **User**:* **User** queries **Agent** with unlabeled data and retrieve respective final results from **Agent**.

TABLE I
System Latent Variables and Hyper-Parameters

Notation	Description
N	The number of **DO**s
L	The number of types of labels
$\mathcal{X}$	The feature space, $\mathcal{X} = \mathbb{R}^p$
$\mathcal{Y}$	The label space, $\mathcal{Y} = \{1, \ldots, L\}$
D_i	Labeled dataset owned by i-th **DO**,$i \in \{1, \ldots, N\}$
M_i	The weak model trained with D_i. $M_i(\bullet) : \mathcal{X} \to \mathbb{R}^L$
D^A	Labeled dataset owned by **Agent**
D^Q	Unlabeled dataset queried by **User**
n^A	$\|D^A\|$
n^Q	$\|D^Q\|$
$(\mathbf{x}_t^A, y_t^A)$	The t-th record in D^A, $t \in \{1, \ldots, n^A\}$, where $\mathbf{x}_t^A \in \mathcal{X}$, $y_t^A \in \mathcal{Y}$
$\mathbf{x}_j^Q$	The j-th record in D^Q, $j \in \{1, \ldots, n^Q\}$, where $\mathbf{x}_j^Q \in \mathcal{X}$
$Result_j$	Inference label result for $\mathbf{x}_j^Q \in D^Q$,$Result_j \in \mathcal{Y}$
$Prob_{i,t}^A$	The probabilities distribution on $\mathcal{Y}$ for $\mathbf{x}_t^A \in D^A$ predicted by M_i, $Prob_{i,t}^A \in \mathbb{R}^L$
$Prob_{i,j}^Q$	The probabilities distribution on $\mathcal{Y}$ for $\mathbf{x}_j^Q \in D^Q$ predicted by M_i, $Prob_{i,t}^Q \in \mathbb{R}^L$
$\mathbf{W}^{(\tau)}$	The weights matrix computed by **Agent** at τ-th iteration, $\mathbf{W}^{(\tau)} \in \mathbb{R}^N \times \mathbb{R}^L$
$Pred_t^{A(\tau)}$	The weighted probabilities distribution on $\mathcal{Y}$ for $\mathbf{x}_t^A \in D^A$ at τ-th iteration, $Pred_t^{A(\tau)} \in \mathbb{R}^L$
$Pred_j^{Q(\tau)}$	The weighted probabilities distribution on $\mathcal{Y}$ for $\mathbf{x}_j^Q \in D^Q$ at τ-th iteration, $Pred_j^{Q(\tau)} \in \mathbb{R}^L$
$Dist_j^{(\tau)}$	The estimated probabilities distribution on $\mathcal{Y}$ for $\mathbf{x}_j^Q \in D^Q$ aggregated at τ-th iteration, $Dist_j^{(\tau)} \in \mathbb{R}^L$
Ω	The total iteration times for SC-Agg
α	$\alpha \in (0, 1)$ is the threshold for **Agent** determining whether the prediction is confident
ρ	$\rho \in (0, 1)$ is a probability decay parameter

IV. Problem Formulation

In this section, we formally introduce the aggregation problem that the SC-Agg algorithm targets. Table I lists frequently used system latent variables and hyper-parameters with respective descriptions.

There are N **DO**s in our system. For the i-th **DO**, where $i \in \{1, \ldots, N\}$, it has a small labeled dataset D_i and trains a model M_i with D_i. **User** queries **Agent** with an unlabeled dataset $D^Q = \{\mathbf{x}_j^Q | \mathbf{x}_j^Q \in \mathcal{X}, j \in \{1, \ldots, n^Q\}\}$. **Agent** holds a small labeled dataset $D^A = \{(\mathbf{x}_t^A, y_t^A) | \mathbf{x}_t^A \in \mathcal{X},\ y_t^A \in \mathcal{Y}, t \in \{1, \ldots, n^A\}\}$.

For better description, we assume that y_j^Q is the ground truth label of the $x_j^Q \in D^Q$, and $\{(x_j^Q, y_j^Q) | x_j^Q \in D^Q, j \in \{1, \ldots, n^Q\}\} \sim \mathcal{D}^Q$. Our paper is based on the assumption that,

- neither **Agent** nor **DO**s can predict the distribution of upcoming queried data D^Q until they receive D^Q.
  For any non-linear function $\mathcal{F}$ predicting the distribution of the coming D^Q, we have
  $$\mathbf{pr}(\mathcal{F}(D^A, \{D_1, \ldots, D_N\}) = \mathcal{D}^Q) = \epsilon, \quad (1)$$
  where ϵ is negligible.

On receiving the queried D^Q from **User**, **Agent** forwards D^Q to remote models and retrieves responses for aggregation. By iteratively excuting SC-Agg for Ω times, **Agent** aggregates out the final $Result$ with

$$Result = \arg\max(Pred^{Q(\Omega)}), \quad (2)$$

where $Pred^{Q(\Omega)}$ is the weighted probabilities distribution for D^Q calculated by **Agent** after Ω times iteration of SC-Agg.

We aim at generating high accuracy responses for the outside queries under limited prior knowledge. We measure the performance of our system with

$$\mathcal{Q} = \mathbf{pr}((x_j^Q, Result_j) \sim \mathcal{D}^Q | D^A, D^Q, \{M_1, \ldots, M_N\}). \quad (3)$$

The goal of our system is to elevate $\mathcal{Q}$, *i.e.*, predict queries in higher accuracy. For $Result$ being the $\arg\max$ result of $Pred^{Q(\Omega)}$, the fundamental problem for SC-Agg to solve is how to compute the weighted probabilities distribution $Pred^{Q(\Omega)}$ precisely such that the $\arg\max$ for $Pred^{Q(\Omega)}$ approximates to the ground truth labels of D^Q.

V. Methodology

As described in Section III, only three roles are involved in our system, *i.e.*, the **DO**s, the **User**, and the **Agent**. The i-th **DO** privately trains M_i with a small sensitive dataset D_i. **DO** answers **Agent** with the predicted probabilities distributions $Prob^Q$ (or $Prob^A$) for the queried data D^Q (or D^A leaving out labels). **User** queries **Agent** with an unlabeled dataset D^Q, and receives respective labels $Result$ from **Agent**.

To achieve high accuracy in answering queries securely and efficiently, we introduce the light-weight Secure Crowdsourcing Aggregation (SC-Agg) algorithm. Referring to Section IV, the essential element in aggregating high quality $Result$ is to precisely compute the weighted probabilities distribution $Pred^{Q(\Omega)}$ for higher accuracy of $Results$. In this section, we describe how SC-Agg fulfills this goal in detail.

A. Secure Crowdsourcing Aggregation Algorithm (SC-Agg)

According to our system settings, SC-Agg is iteratively executed Ω times for gradually generating accurate final $Result$ toward a queried D^Q. In this section, we specify the process of the τ-th $(0 < \tau \leq \Omega)$ execution of SC-Agg.

Referring to Section IV, the motivation of SC-Agg is refining $Pred^{Q(\tau-1)}$. Intuitively, the content of $Pred^{Q(\tau)}$ is related to the $Pred^{Q(\tau-1)}$, a weights matrix, and the predict probabilities distribution for D^Q collected from distributed models. Mathematically, we have

$$Pred^{Q(\tau)} \leftarrow \texttt{softmax}(Pred^{Q(\tau-1)} + \sum_{i=1}^{N} Prob_i^Q \cdot \mathbf{W}_i^{(\tau)}). \quad (4)$$

As $Prob^Q$ collected from distributed models is fixed for one D^Q, the potential factor impacting contents of $Pred^{Q(\tau)}$ is a series of weights matrixes $\{\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(\tau)}\}$. $\mathbf{W}^{(\tau)}$ is a models capacities reflecting matrix designed for emphasizing strengths and weakening shortages for models. For example, at the τ-th analysis, there is a $M_{\hat{i}}$ predicting label l_1's classification tasks with high accuracy, and performing weak in classifying label l_2. To stress the advantage of $M_{\hat{i}}$ in label l_1's classification tasks, the value $\mathbf{W}_{\hat{i},l_1}^{(\tau)}$ should be fostered. Conversely, $\mathbf{W}_{\hat{i},l_2}^{(\tau)}$ is relatively lower in order to reduce the negative impact brought from $M_{\hat{i}}$.

In machine learning, when a classifier's inference capacities are distributed similar to the queried data distributions, it shows higher accuracy in this inference task. Therefore, computing $\mathbf{W}^{(\tau)}$ is related to $\{M_1, \ldots, M_N\}$ inference capacities and D^Q's distribution. Due to the lack of prior experience about D^Q's distribution, in SC-Agg, we introduce $Dist^{(\tau)}$ as the estimated distribution of queried D^Q after τ times of iterations. Thus, each time before computing $\mathbf{W}^{(\tau)}$, SC-Agg updates the estimation for D^Q's distribution $Dist^{(\tau)}$.

Roughly speaking, in the τ-th round of SC-Agg, **Agent** sequently updates $Dist^{(\tau)}$, $\mathbf{W}^{(\tau)}$, and $Pred^{Q(\tau)}$ as shown below.

- *$Dist^{(\tau)}$:* Computing estimated distributions $Dist^{(\tau)}$ involves two hyper-parameters, *i.e.*, α and ρ. $\alpha \in (0, 1)$ is the minimum probability value for **Agent** judging the prediction is confident. $\rho \in (0, 1)$ is a probability decay parameter to determine the probability for replacing the estimated probabilities distribution to weighted one. At the τ-th iteration, for a record $x_j^Q \in D^Q$, if the $\max(Dist_{j,*}^{(\tau-1)}) < \alpha$, then **Agent** believes the estimated probabilities distribution $Dist_j^{(\tau-1)}$ for x_j^Q is not confident. For a more accurate estimation, **Agent** replaces all uncertain distributions in $Dist^{(\tau-1)}$ to the respective ones in $Pred^{Q(\tau-1)}$ with the probability $1 - \rho^\tau$. As the τ increases over time, $Pred^{Q(\tau-1)}$ is more stable and accurate, and thus the replacing probability $1 - \rho^\tau$ is increasing for better estimation in $Dist^{(\tau)}$. After the analysis and replacements, $Dist^{(\tau-1)}$ is updated to $Dist^{(\tau)}$.
- *$\mathbf{W}^{(\tau)}$:* $\mathbf{W}^{(\tau)}$ is the most essential parameter possessed by **Agent** for it determining the output results directly. To

obtain a $\mathbf{W}^{(\tau)}$ with higher probability in computing out accurate inference results, we update $\mathbf{W}^{(\tau-1)}$ based on the correlation between the estimated D^Q's distribution and the current weights matrix $\mathbf{W}^{(\tau-1)}$.
As a reflection of capabilities for $\{M_1, \ldots, M_N\}$ in predicting D^Q, higher $\mathbf{W}_{i,l}^{(\tau)}$ implies higher accuracy for M_i classifying data in type l after τ times iterated analysis. Because D^Q and all $\mathbf{W}_i^{(\tau)}$ are not distributed in a normal distribution, we choose Spearman's rank correlation coefficient [15] to reflect the relationship between $Dist^{(\tau)}$ and every **DO**'s estimated inference capacity, *i.e.*, $\mathbf{W}_i^{(\tau)}$. The larger Spearman's rank correlation coefficient, the more likely a positive correlation between $Dist_l^{(\tau)}$ and $\mathbf{W}_{i,l}^{(\tau)}$, and thus more possibility the M_i provides a correct answer for the queried data in type l. We regularize Spearman's rank correlation coefficient cor_i to [0,1) by substituting the cor_i with $\max(0, cor_i)$. Then we compute $\mathbf{W}_i^{(\tau)}$ with

$$\mathbf{W}_i^{(\tau)} \leftarrow \frac{1}{\mathbf{Z}^{(\tau)}} \mathbf{W}_i^{(\tau-1)} \cdot \exp(-cor_i / \sum_{b=1}^{N} cor_b), \tag{5}$$

where $\mathbf{Z}^{(\tau)}$ is a regularized parameter to emphasis the relative capabilities of each **DO**. After regularization, $\forall l \in \mathcal{Y}$, $\sum_{b=1}^{N} \mathbf{W}_{b,l}^{(\tau)} = 1$.
To explore more possibilities in updating $\mathbf{W}^{(\tau)}$, we introduce the local search method. We construct a $\mathbf{W}^r$ by applying a little random disturbance to the original $\mathbf{W}^{(\tau)}$. We decide whether to substitute $\mathbf{W}^{(\tau)}$ with $\mathbf{W}^r$ with their estimated quality. To estimated quality of $\mathbf{W}^r$ and $\mathbf{W}^{(\tau)}$, from D^A, **Agent** first extracts a subset D' which has the same distribution as $Dist^{(\tau)}$. (6) shows the constrains for the subset D'.

$$\frac{|\{\mathbf{x}_t^A | (\mathbf{x}_t^A, l) \in D'\}|}{|\{j | l = \arg\max_{\hat{l}}(Dist_{j,\hat{l}}), j \in \{1, \ldots, n^Q\}\}|} = \mathcal{C}, \tag{6}$$

where $\mathcal{C}$ is a constant to determine $|D'|$, $\forall l \in \{1, \ldots, L\}$. Then we estimate the quality of tested weights matrix by computing the inference accuracy on D' separately by

$$\mathcal{Q}^{\mathbf{W}} \leftarrow \mathbf{pr}(\arg\max(Prob_t^A \cdot \mathbf{W}) = y_t | (\mathbf{x}_t, y_t) \in D', Prob^A), \tag{7}$$

where $\mathbf{W}$ can be either $\mathbf{W}^r$ or $\mathbf{W}^{(\tau)}$.
After computing out estimated quality of $\mathbf{W}^r$ and $\mathbf{W}^{(\tau)}$, **Agent** picks the one with higher estimated quality for next iteration.

- $Pred^{Q(\tau)}$: Computing $Pred^{Q(\tau)}$ with (4).

In SC-Agg, for the τ-th iteration, **Agent** first computes the latest estimated probabilities distributions $Dist^{(\tau)}$ with $Dist^{(\tau-1)}$ and $Pred^{Q(\tau-1)}$. Then **Agent** relies on its own estimated distributions $Dist^{(\tau)}$ and $\mathbf{W}^{(\tau-1)}$ to get the up-to-date weights matrix $\mathbf{W}^{(\tau)}$. During the process of computing $\mathbf{W}^{(\tau)}$, **Agent** introduces the random local search [16] for more possibilities in promoting the estimated accuracy of queries. With $\mathbf{W}^{(\tau)}$ settled, **Agent** updates the weighted probabilities distributions $Pred^{Q(\tau)}$, and then regularizes it using softmax. By repeatedly updating $Dist$, $\mathbf{W}$, and $Pred^Q$, **Agent** gets the final output $Result$ based on $Pred^{Q(\Omega)}$. The whole SC-Agg method is generalized in Algorithm 1.

Algorithm 1: SC-Agg

```
Input: $\tau, \rho, \alpha, D^A, D^Q, Prob^A, Prob^Q, Dist^{(\tau-1)}, \mathbf{W}^{(\tau-1)}, Pred^{Q(\tau-1)}$
Output: $Dist^{(\tau)}$, $\mathbf{W}^{(\tau)}$, $Pred^{Q(\tau)}$
1  foreach $j \in \{1, \ldots, n^Q\}$ do
2      if $\max(Dist_{j,*}^{(\tau-1)}) < \alpha$ then
3          With probability $1 - \rho^\tau$,
             $Dist_j^{(\tau-1)} \leftarrow Pred_j^{Q(\tau-1)}$
4      end
5  end
6  $Dist^{(\tau)} \leftarrow Dist^{(\tau-1)}$;
7  foreach $i \in \{1, 2, \ldots, N\}$ do
8      Computing Spearman's rank correlation
         coefficience $cor_i$ between $\mathbf{W}_i^{(\tau-1)}$ and $Dist^{(\tau)}$ ;
9      Computing $\mathbf{W}_i^{(\tau)}$ with (5);
10 end
11 Randomly generate a small disturbing matrix
     $\mathbf{W}^{dis} \in \mathbb{R}^N \times \mathbb{R}^L$;
12 $\mathbf{W}^r \leftarrow \mathbf{W}^{(\tau)} + \mathbf{W}^{dis}$;
13 Estimating quality of $\mathbf{W}^r$ and $\mathbf{W}^{(\tau)}$ with (7);
14 Choosing the weights matrix with higher estimated
     quality to be the $\mathbf{W}^{(\tau)}$;
15 Update $Pred^{Q(\tau)}$ with (4)
```

B. Inference

The overall inference process can be divided into two steps, 1) the initialization step, and 2) the aggregation step. The initialization step is the preparation step for the upcoming aggregation process. The aggregation step is mainly for **Agent** to perform Ω times of SC-Agg then finally output the results back to the **User**.

In the initialization step, **Agent** prepares $\mathbf{W}^{(0)}$, $Dist^{(0)}$, and $Pred^{Q(0)}$ for executing SC-Agg. Before starting the computation, **Agent** receives the queried data D^Q, $Prob^A$ and $Prob^Q$ as the predict probabilities distributions from the models ensemble for D^A and D^Q respectively. The details are given below.

1) **Agent** initializes the weights matrix $\mathbf{W}^{(0)}$ by evaluating each M_i's capacity in classifying label l with

$$\mathbf{W}_{i,l}^{(0)} \leftarrow \frac{1}{\mathbf{Z}^{(0)}} \mathbf{pr}(\arg\max M_i(\mathbf{x}) = l | M_i, (\mathbf{x}, l) \in D^A), \tag{8}$$

where $i \in \{1, \ldots, N\}$, and $\mathbf{Z}^{(0)}$ is a regularized parameter. After regularization, $\forall l \in \mathcal{Y}$, $\sum_{b=1}^{N} \mathbf{W}_{b,l}^{(0)} = 1$.

2) For $Dist^{(0)}$ initialization, **Agent** fills $Dist^{(0)}$ with

$$Dist_j^{(0)} \leftarrow \texttt{softmax}(||\mathbf{x}_j^Q - E(\mathbf{x}|(\mathbf{x}, 1) \in D^A)||_2, \ldots, ||\mathbf{x}_j^Q - E(\mathbf{x}|(\mathbf{x}, L) \in D^A)||_2) \tag{9}$$

where $j \in \{1, \ldots, n^Q\}$, and the $E(\cdot)$ means the expectation of the inner parameter.

3) On collecting $Prob^Q$ and initializing the weights matrix $\mathbf{W}^{(0)}$, **Agent** fills the $Pred^{Q(0)}$ with

$$Pred_j^{Q(0)} \leftarrow \texttt{softmax}(\sum_{i=1}^{N} Prob_{i,j}^{Q} \cdot \mathbf{W}_i^{(0)}), \qquad (10)$$

where $j \in \{1, \ldots, n^Q\}$.

After the initialization, **Agent** iteratively performs SC-Agg Ω times. Finally, **Agent** returns aggregated $Result$ from the weighted probabilities distributions $Pred^{Q(\Omega)}$ with (2).

C. Complexity Analysis

In this section, we analyze the communication complexity, the amount of transmitted data, and the time complexity in our system. From the description above, our system initializes once, and whenever D^Q arrives, our system performs Ω times of SC-Agg to output $Result$. Thus, our analysis for system complexity is divided into the system initialization step and the aggregation step.

- **Communication Complexity:** At the system initialization step, **Agent** queries **DO**s with D^A and **Agent** receives the probabilities distributions from **DO**s. The communication complexity at this step is $\mathcal{O}(N)$. Each time **User** queries **Agent** with D^Q, **Agent** queries **DO**s with D^Q and receives the probabilities distributions from **DO**s. The communication complexity at the querying and answering step is also $\mathcal{O}(N)$.
- **Amount of Data Transmitted:** The amount of data being transmitted in our system is determined by the number of **DO**s, the size of D^A, the size of queried D^Q, and the labels space L. At the system initialization step, the amount of transmitted data is $\mathcal{O}(N * n^A * (L+p))$. When **Agent** processes **User** queried data D^Q, the total amount of transmitted data is $\mathcal{O}(N * n^Q * ((L+p)+p+1))$.
- **Time Complexity:** Time complexity for our system is determined by the number of items to be computed and updated during the process. In the system initialization, the time complexity is $\mathcal{O}(L * (N + n^Q + 1))$. The time complexity from receiving D^Q to perform Ω times SC-Agg process and output $Result$ is $\mathcal{O}(n^Q * L + \Omega * L * (n^Q + N + n^A) + n^Q)$.

Updating models' parameters in general FML methods require much more time than inference [17]. Comparing to the high complexity in communication for exchanging training information in [8], we only require $\mathcal{O}(N)$ communication complexity for the whole aggregation process. Surpassing [9] that also targets in light-weight privacy-preserving distributed IoT system, SC-Agg saves expensive encryption computation for end devices and eliminates millions of FLOP operations for the central server to train a new model. The amount of data to be transmitted in one-time communication for aggregating is also much less than the generic FML process in which a large number of floating-point matrices need to be exchanged.

VI. Experiments

In this section, we experimentally verify that SC-Agg algorithm performs well in different scenarios. We compare the performance of SC-Agg with two baselines[1] , 1) the majority vote (MV); 2) the best model among all **DO**s (BM). We use Δ_1 to denote how much SC-Agg increases the average queries accuracy comparing to the BM baseline, and we compute Δ_2 as the increment SC-Agg brings to the MV baseline. Our experiments are executed on the MNIST [18] real-world data collections. For all testing scenarios, the proposed algorithm SC-Agg improves the baselines significantly.

The inference process involves three phases, 1) estimating the distribution of queried data, 2) analyzing the answers from **DO**s, and 3) iteratively refining a weights matrix according to **Agent**'s dataset and the queried data. The performance of our system is mainly affected by three factors, the size of D^A, the number of **DO**s, and the average performance of **DO**s. From our results, SC-Agg performs consistently in all testing situations.

A. Experimental Configuration

We conduct a series of experiments on the MNIST dataset with 5, 10, 20, 50, and 100 **DO**s. MNIST is a database that contains handwritten digits classified from 0 to 9. It has a training set of 60,000 examples and a test set of 10,000 examples. To fulfill system settings in Section III that the D^A owned by **Agent** are not sensitive related, we select two subsets without overlap from the training examples as the sensitive training data and D^A separately. In the sensitive dataset, we randomly select N subsets and distributed to every **DO**, note that N is the number of **DO**s. Recall our assumptions that these N sensitive data subsets are distributed differently. Thus, distributed models trained by these subsets have diversified capacities for different inference tasks. We train each **DO**'s model using the respective sensitive dataset with a Convolutional Neural Network in the same structure and implement SC-Agg with Python 3.6.0. Our experiments are executed on one GeForce GTX 1080 GPU, four Intel(R) Xeon(R) Silver 4108 CPU @ 1.80GHz and 28 GB RAM. We refer to the model inference accuracy on the whole MNIST test set as the "testing accuracy". Without loss of generality, we design several D^Qs with random distributions and in different sizes. To control the variance, the contents of these D^Qs are constant in every experiment. Specifically, we query our system 16 times with different sizes of D^Q. For the x-th queried dataset, where $x \in \{1, 2, \ldots, 16\}$, it is consisted with ($x$*300-100) unlabeled data randomly chosen from MNIST test set. Our experimental results show that SC-Agg outperforms the other two baselines consistently regardless of the varied distributions nor the sizes of queried D^Q.

B. Experimental Results and Analysis

Firstly, We test different n^A on 50 **DO**s to verify that, with the increment of n^A, the performance of the whole system is getting better. Our n^As are set to 150, 300, and 600, and the experimental results are shown in Table II. From Table II, we find out that as n^A increases, the aggregation accuracy for

[1]To have a fair comparison, we do not compare our approach with the ones that have privacy leakage such as some existing FML solutions

TABLE II
EXPERIMENTAL RESULTS OF 50 **DO**S WITH DIFFERENT n^A

n^A	**BM**	**MV**	**SC-Agg**	Δ_1	Δ_2
150	64.72%	74.76%	**78.06%**	13.33%	3.30%
300	64.72%	74.76%	**78.24%**	13.52%	3.49%
600	64.72%	74.76%	**78.31%**	**13.58%**	**3.55%**

TABLE III
EXPERIMENTAL RESULTS OF 5, 10, 20 AND 50 **DO**S

N	**BM**	**MV**	**SC-Agg**	Δ_1	Δ_2
5	64.72%	64.13%	**67.76%**	3.04%	3.63%
10	64.72%	64.61%	**67.33%**	2.61%	2.72%
20	64.72%	68.17%	**72.00%**	7.28%	**3.83%**
50	64.72%	74.76 %	**78.31%**	**13.58%**	3.55%
100	64.72%	75.74%	**78.19%**	13.47%	2.45%

queries improves, which matches our expectations. We infer that the increment of n^A provides our system higher precision in estimating respective model capacity in **DO**s ensemble. Because we estimate possible weights matrix's quality in SC-Agg by computing the respective inference accuracy in a subset of D^A, as in (7), when the n^A is larger, we can choose a larger subset for more a precise simulation on the distribution of queried data. More precise in distribution estimation brings about higher possibilities in measuring a better weights matrix, which finally benefits the output $Result$. As shown in Table II, when n^A is set to 600, SC-Agg boosts the accuracy in answering queries as much as 13.58% comparing to the best model among 50 models, and SC-Agg achieves 3.55% higher than majority voting too. Furthermore, though the size of D^A is as little as 150, *i.e.* with only averagely 15 data records per class, it is sufficient for our system to achieve a better performance compared to the other two baselines. This characteristic of SC-Agg makes it perfect for the scenario where every entity only has limited data. We draw the conclusion that a larger size of D^As produces better results of SC-Agg, while SC-Agg does not require a large dataset for **Agent** to accurately output answers for queried data.

Next, to test how the number of **DO**s, *i.e.*, N, affect SC-Agg's performance, we randomly choose 5, 10, 20, 50, and 100 models with averagely 39.23% testing accuracy to carry out the experiments. The results are listed in Table III. As shown in Table III, SC-Agg outperforms the MV and the BM baselines in all queries regardless of the number of **DO**s involved. Because one of the essential steps in SC-Agg is to estimate the capacities of every model and then assign values to the weights matrix according to the relative capacities (5), the more **DO**s we have in our system, the more accurate assignment for weights matrix **W** we can have. Due to the diversified inference abilities of models in different types of tasks, larger N brings a higher chance of having specialists in one specific type of task. Consequently, as N increases, SC-Agg obtains higher accuracy in answering the queried data. Because SC-Agg considers both the models ensemble and the queried data to generate the final output, even when there are only 5 weak classifiers, SC-Agg surprisingly outperforms the MV baseline of 10 **DO**s, and with less than 1% accuracy loss compared to the MV baseline in 20 **DO**s scenarios. It is worth to point out that, as the number of **DO**s increases, the improvement of SC-Agg over BM increases. For all tested numbers of **DO**s, SC-Agg consistently outperforms the MV baseline by 3.24% on average. Also, SC-Agg achieves the best queried accuracy when there are 50 **DO**s, *i.e.* 78.31%, with 13.58% higher than the BM baseline.

TABLE IV
EXPERIMENTAL RESULTS OF 5 **DO**S IN DIFFERENT AVERAGE CAPACITIES

Average Testing Accuracy	**BM**	**MV**	**SC-Agg**	Δ_1	Δ_2
81.26%	87.92%	89.47%	**89.79%**	1.87%	0.32%
61.59%	74.66%	77.45%	**82.12%**	7.47%	4.67%
44.79%	64.72%	64.13%	**67.76%**	3.04%	3.63%
31.09%	37.64%	45.02%	**50.32%**	**12.68%**	**5.31%**

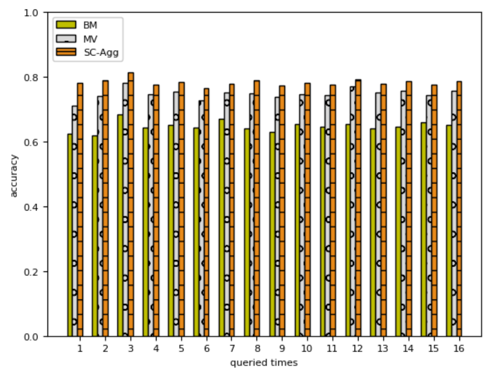

Fig. 2. Experiments for 50 **DO**s, $n^A = 300$

Finally, we test SC-Agg performances with **DO**s of different average prediction abilities. In this experiments, we compare the outputs of 5 **DO**s with different average testing accuracy, *i.e.* 81.26%, 61.59%, 44.79%, and 31.09%. Table IV shows the respective experimental results. Referring to Table IV, SC-Agg can consistently outperform the other two baselines in all situations. Though when the models' average testing accuracy is as high as 81.26%, which violates the assumption for our problem that each model is weak, SC-Agg still shows steady improvement compared to the other two baselines. When **DO**s only have weak classifiers, *i.e.* all models are in low average testing accuracy, SC-Agg can have significant improvement compared to both the MV baseline and the BM baseline. As shown in Table IV, when the average testing accuracy is 31.09%, SC-Agg elevates the BM baseline as high as 12.68%, which is almost half of the BM accuracy.

C. Experimental Summary

Table II lists the experimental results for testing how the number of **DO**s affects the performance of SC-Agg. Table III shows the results in studying how D^A's size impacts SC-Agg. Table IV lists the results for experiments on **DO**s with different average inference capacities.

According to our experimental results, SC-Agg achieves higher accuracy in answering queried data than both the BM baseline and the MV baseline in all tested scenarios. We also show that **Agent** only needs a small number of labeled data for SC-Agg to perform well. We believe that our method is applicable for real-world applications, especially when we need to assemble of weak classifiers. Referring to Table III, SC-Agg boosts the prediction accuracy from 64.72% to 78.31% compared to the highest queried accuracy among 50 weak models. Furthermore, in Table IV, when only as little as five weak classifiers are available, SC-Agg accomplishes 5.31% higher in queried accuracy than majority voting. Note that SC-Agg contains three hyper-parameters, *i.e.* Ω, α and ρ. For different numbers of **DO**s and various types of inference tasks, we need to tune our hyper-parameters values accordingly to attain good performance. As D^A is an essential dataset for SC-Agg achieving notable performance, through inspecting our experimental results, **Agent** with D^A containing data averagely distributed in each category can aggregate out $Result$ in higher accuracy.

VII. Conclusions

Our paper considers the problem of how to aggregate the inference results of individual IoT machine learning models for better overall inference performance while avoiding privacy leakage. Based on Federated Machine Learning (FML), we rely on a trusted central server to construct a Privacy-Preserving Federated Learning System. In our system, the agent only collects the inference results from individual IoT devices. Our system requires no sensitive information on remote individual learning models for minimizing the possibility of privacy attacks. Under this framework, we also proposed a novel Secure Crowdsourcing Aggregation Algorithm (SC-Agg) for our system that can provide more accurate predictions based on diversified inference results from individual IoT devices and does not rely on historical information of queries, unlike traditional crowdsourcing algorithms. Our experimental results for different scenarios show that SC-Agg outperforms the majority voting and the best performance among all data owners. According to our experimental results, SC-Agg not only improves the MV baseline by at most 5.31% (Table IV) but also is capable of achieving as high as 13.58% increment for BM baseline (Table III). In summary, we believe that SC-Agg can be used in real-word distributed IoT systems.

On the other hand, our system has some accuracy loss comparing to the model trained with all used sensitive data. As a future direction, we will focus on designing how data owners communicating with each other and the trusted third party securely to reduce the accuracy gap between our system with the model trained with all data in the system.

Acknowledgements

This project is partially supported by the CRF grant of HKSAR (City U: C1008-16G).

References

[1] J. Konečný, H. B. Mcmahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016.

[2] J. Konečný, H. B. Mcmahan, F. X. Yu, P. Richtárik, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016.

[3] H. B. Mcmahan, E. Moore, D. Ramage, and B. A. Y. Arcas, "Federated learning of deep networks using model averaging," 2016.

[4] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 739–753.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[6] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018.

[7] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *CoRR*, vol. abs/1906.08935, 2019. [Online]. Available: http://arxiv.org/abs/1906.08935

[8] L. Jiang, R. Tan, X. Lou, and G. Lin, "On lightweight privacy-preserving collaborative learning for internet-of-things objects," 04 2019.

[9] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu, "Deep learning towards mobile applications," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1385–1393.

[10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," 2018.

[11] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 491–518, 2012.

[12] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *International Conference on Neural Information Processing Systems*, 2011.

[13] K. Mo, E. Zhong, and Y. Qiang, "Cross-task crowdsourcing," in *Acm Sigkdd International Conference on Knowledge Discovery & Data Mining*, 2013.

[14] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2016.

[15] L. Myers and M. J. Sirois, "Spearman correlation coefficients, differences between," *Encyclopedia of statistical sciences*, vol. 12, 2004.

[16] B. Selman, H. A. Kautz, and B. Cohen, "Noise strategies for improving local search," in *AAAI*, vol. 94, 1994, pp. 337–343.

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

[18] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/