A Privacy Preserving Federated Learning Framework for COVID-19 Vulnerability Map Construction

Jeffrey Jiarui Chen*, Rui Chen†, Xinyue Zhang† and Miao Pan†
*St. Mark's School Of Texas, 10600 Preston Rd, Dallas, TX 75230
†Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500975

***Abstract*—This paper presents a federated learning (FL) framework that uses multiple self-reporting crowdsourcing mobile and web apps to collaboratively construct a fine-grained COVID-19 vulnerability prediction map. The use of FL provides a reliable prediction by aggregating training results from multiple apps, while at the same time circumventing data privacy regulations that prevent user information from multiple apps to be shared with each other. Such a fine-grained vulnerability map identifies early on high-risk areas, helping to reduce the spread of the disease. To mitigate data bias from each self-reporting app, an adaptive worker selection algorithm that leverages neighbouring datasets to obtain a balanced data distribution is proposed. Further, a differential privacy scheme is adopted to protect user information. The simulation results show that the proposed framework outperforms the widely used FedAvg FL algorithm by 6% on prediction accuracy while preserving user privacy.**

I. Introduction

The recent COVID-19 pandemic has caused a public health crisis around the world. It is critical to identify early on high-risk areas and to forecast future cases, which is also called vulnerability risk. COVID-19 heatmaps that show the locations of people with high risk being infected have helped the public understand COVID-19 transmission in communities. The heatmaps also allow healthcare organizations to proactively allocate medical resources to stop the virus from spreading.

The success of the COVID-19 vulnerability map construction relies on comprehensive COVID-19 data. The maps are normally generated based on infection information from local governments or the Centers for Disease Control and Prevention (CDC). However, current vulnerability maps fail to provide reliable and detailed information. Existing maps only show confirmed cases at the county level. They don't provide fine-grained levels of vulnerability and lack of adequate coverage of patients who are asymptomatic. Consequently, there has been an increase in the number of various mobile and web self-reporting applications (apps) that report crowdsourced symptom data. Such crowd-sourcing apps collect a tremendous amount of data tagged with specific geographic information and play a major role in monitoring COVID-19. It becomes promising to construct a fine-grained vulnerability map leveraging data from multiple apps to reliably predict vulnerability [1].

Recently, machine learning (ML) models have been used to construct COVID-19 maps that predict the future trend of the disease. Nevertheless, using only a single app to collect the user dataset to train the ML models has drawbacks. Data collected from a single app represents a limited geographic community or a particular group. One promising solution is to make the predictions more reliable through collaboration among multiple crowdsourcing apps. However this approach raises several new challenges. The first is the privacy concern. Crowdsourced symptom related data contains sensitive information and transferring this information is regulated by the government. This regulation prevents data from multiple crowdsourcing apps to be shared. Another issue is that the crowdsourcing data may not distribute uniformly among devices, which occurs when the users of an app are from a particular community and represent a similar demographic. This could lead to misrepresentation of certain groups, potentially generating an inaccurate map. For instance, data collected from an app that is used by senior citizens could contain more infected people than those from an app predominantly used by school students. Consequently, if the apps used by seniors have more users, the prediction results become biased toward the seniors, which makes the prediction unreliable [2]. Last but not least, in some rural areas, there may be not enough crowdsourcing data collected and the small dataset could cause the overfitting problem in ML models. For example, Facebook has created a COVID-19 interactive map. However, since it only uses one app to collect information, many areas either do not have data or do not have sufficient data to make accurate representations of the COVID-19 spread [3].

To overcome the issues above, we propose to develop a publicly-available federated learning (FL) framework that enables multiple self-reporting apps to cooperate with each other. FL is a state-of-the-art ML approach that seeks to address the problems of data governance and privacy by training algorithms collaboratively without exchanging raw input data [4] [5]. Most recently, several approaches utilizing FL have been developed for COVID-19 applications, such as using X-ray image analysis to detect lung infections [6]. To the best of our knowledge, our approach using FL method to construct a reliable and accurate COVID-19 vulnerability map is unique. The framework contains a central server and local ML models running on individual mobile or web app

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

providers. A long-short term memory (LSTM) model [7] is utilized to train the ML model in our framework. To mitigate the biased training data distribution in FL, we propose an adaptive worker selection algorithm that leverages the training results from selected workers in certain neighborhoods (a worker is defined as an individual crowdsourcing app provider). The algorithm reduces the prediction bias introduced by an underrepresented local training dataset. It also helps mitigate the overfitting problem due to the relatively small datasets. To protect user information, a differential privacy scheme (DP) [8] is utilized in the adaptive worker selection algorithm. Extensive simulations are conducted based on publicly-available datasets to evaluate the performance of the proposed framework. The results demonstrate that the proposed framework not only outperforms the state-of-the-art FL FedAvg algorithm by 6% on prediction accuracy, but also preserves user privacy.

II. System Model & Privacy Preliminary

A. System Model

The proposed FL framework coordinates the collaboration of multiple COVID-19 self-reporting apps to construct a fine-grained and periodically-updated vulnerability prediction map (VPM), as shown in Fig. 1. More specifically, in the VPM, the targeted area $\mathcal{G}$ is divided into K non-overlapping cells, denoted by the set $\mathcal{G} = \{g_1, \cdots, g_k, \cdots, g_K\}$. The FL framework contains a central server and many app providers. Furthermore, the central server creates a broker for each cell and manages the broker operations. There are K number of brokers, denoted by $\mathcal{B} = \{b_1, \cdots, b_k, \cdots, b_K\}$.

For the k-th cell g_k, the broker b_k is responsible for performing an area vulnerability prediction via FL. The central server supervises the crowdsourcing apps as they join the framework. An app is allowed to join the framework if it contains user-reported COVID-19 symptoms data and provides the central server with a list of its workers. A worker is denoted as d and it collects and stores self-reporting data. The workers can be from different apps and each worker can store user reports gathered from more than one cell. The central server sends the workers' IDs to the corresponding broker. The workers available in the b_k broker are denoted as $\mathcal{D}_k = \{d_k^1, \cdots, d_k^m, \cdots, d_k^{M_k}\}$, where d_k^m denotes the m-th worker in cell g_k and M_k is the number of participating workers in cell g_k. The broker b_k coordinates the workers $\mathcal{D}_k$ to jointly train the FL model. A broker doesn't know about any user data information since such information is stored locally on each worker and won't be shared to the broker during training. Further to mitigate bias, we propose an adaptive worker selection algorithm, shown in section IV. After training, the broker b_k sends the prediction results to the central server to build the vulnerability map.

The COVID-19 symptom data is collected by COVID-19 crowdsourcing apps. Each app launches an online questionnaire that asks users which COVID-like symptoms they have along with their location information. We assume the users trust the app to protect their private information and accurately report their symptoms. The app's users are distributed over the region $\mathcal{G}$. At each period, a total of N participants upload their records. We denote the report stored in the j-th app device from the i-th participant as $\boldsymbol{r}_i^j = (t_i^j, \boldsymbol{s}_i^j)$, where t_i^j is the recording timestamp and $\boldsymbol{s}_i^j = (s_{i1}^j, \cdots, s_{ip}^j)$ represents the symptom information with the p covid-related symptom attributes. Each reported $\boldsymbol{r}_i^j$ is tagged with a location l_i^j. In a fine-grained VPM, the spatial unit is set to the street or township level, such as a zip code. Each cell g_k is tagged with a certain vulnerability prediction level V_k. The entire VPM is modeled as $\boldsymbol{V} \triangleq [V_1, \cdots, V_K]$.

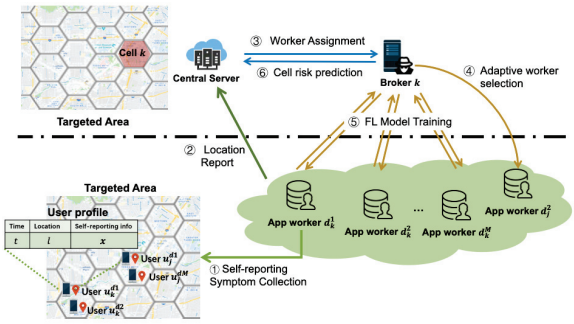

Fig. 1. Federated learning framework for COVID-19 vulnerability map construction.

B. Differential Privacy Preliminaries

The state-of-the-art DP [8] is used as our privacy model in this paper. It gives a rigorous privacy guarantee against what an adversary can infer after observing the published statistics of the users' dataset. The definition of DP is shown as follows.

***Definition** 1:* Suppose privacy parameter $\epsilon \geq 0$, a randomization algorithm $\mathcal{A}$ satisfies ϵ-differential privacy. For any neighboring database x, y that differ in at most one element, and for any subset of outputs $\mathcal{O} \subseteq range(\mathcal{A})$,

$$Pr[\mathcal{A}(x) \in \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{A}(y) \in \mathcal{O}], \tag{1}$$

The different choices of privacy budget ϵ determine different privacy preservation levels. A smaller privacy budget ϵ indicates the probability of the outputs of the randomized algorithm $\mathcal{A}$ with two different datasets is close to each other. This suggests stronger privacy protection. On the other hand, a high privacy preservation level would largely compromise the data utility.

***Definition** 2 (Global sensitivity):* The global sensitivity of a query function f, given two neighboring databases x, y is $\Delta f = \max_{\|x-y\|=1} |x - y|$.

The global sensitivity of a function f represents the maximal difference of the outputs that a single input value changes in calculating the query function f, which only depends on the query function but not the dataset. The sensitivity Δf determines the scale of differential privacy noise needed to efficiently hide the individual information.

In the proposed FL framework, the brokers are assumed to be semi-honest, i.e., honest-but-curious about the users individual information [9] and the workers can only trust

themselves. Thus, secure multi-party computation (SMC) with DP is more suitable in our case. Particularly, we exploit the distributed differential privacy (DDP) framework with Laplace perturbation scheme in [10], where the workers generate partial noise from Laplace distribution. Then the workers inject it to their data locally such that the aggregated Laplace noise is large enough to guarantee DP. Meanwhile a secure aggregation algorithm is introduced to ensure the only information the aggregator (i.e., the broker) can observe is the statistical results, which is shown in Section IV-C.

III. COVID-19 Risk Level Estimation

The first step is the individual risk assessment, in which the collected symptoms are mapped into a vulnerability level. Since the self-reporting questionnaires launched by each app are not exactly the same, we assume that the symptoms contained in the questionnaires are similar to the symptoms provided by the CDC, such as cough, fever with high body temperature, chest pain and shortness of breath. Specifically, given the reported symptoms $\mathcal{X} = \{\boldsymbol{x}_i, 1 \leq i \leq N\}$, the worker would determine the vulnerability of each participant via a predetermined function $h : \mathcal{X} \rightarrow (0, 1)$. The vulnerability score of each user is evaluated as the number of reported symptoms divided by the total number of symptoms in the predefined list. If the participant has tested positive for COVID-19, then $h = 1$. By associating users with grid cells based on locations l, each worker estimates the grid-level vulnerability from the individual estimates, and then the broker aggregates the workers' results as $V_k = \frac{\sum_{i=1}^{n_k} h(\boldsymbol{x}_i)}{N_k}$ where N_k denotes the total population in the grid cell g_k and n_k is the number of participants in the grid.

The questionnaires include the participant's age information, which is used in our training algorithm to reduce bias in the data collected from multiple apps, helping to achieve a balanced representation in the model development. This method will be described in the next section.

IV. Vulnerability Prediction

The use of the standard FL framework for vulnerability prediction could result in a degradation in accuracy because of the imbalance of the datasets from multiple apps. To address this issue, an adaptive worker selection algorithm is proposed and a DDP scheme is utilized to protect the user information. A data pre-processing approach is also exploited to handle the small dataset issue. A LSTM model is used in our FL framework for vulnerability prediction.

A. Adaptive Federated Learning

We start with utilizing the federated averaging (FedAvg) [5] method as a baseline to predict the COVID-19 vulnerability risk. Later, our model will improve on the FedAvg method. In the following description, we use one broker as an example to illustrate our FL process. Each cell has a ML model, called a global model. Each individual worker's ML model is called a local model. A worker can be selected for more than one cell's FL model training process. The broker and the workers encrypt the communication messages containing the model parameters to provide data security.

Algorithm 1 Adaptive Federated Averaging

Require: p_k is the size of workers in cell k and p'_k is the size of workers from the g's neighborhood cells k' for training; E is the size of local epochs; the U_k are the workers in cell k, and the U'_k are the workers in k's neighborhood cells k'.

Ensure: FL model ω

procedure BROKERUPDATE:
Initialize ω_0
  for each round $t = 1, 2, ...$ **do**
    $U_k \leftarrow$ (random p_k workers)
    if (Adaptive Worker Selection) **then**
      $U'_k \leftarrow$ (KLD used to select p'_k workers in neighbor cells k')
    else
      $U'_k \leftarrow 0$
    end if
    for each worker $d \in U_k$ and $d' \in U'_k$ in parallel **do**
      $\omega^d_{t+1} \leftarrow \text{WorkerUpdate}(d, \omega_t, k)$
      $\omega^{dk'}_{t+1} \leftarrow \text{WorkerUpdate}(d', \omega_t, k')$
    end for
    if (Adaptive Worker Selection) **then**
      $\omega_{t+1} \leftarrow \alpha \sum \frac{n^{d_i}_k}{n_k} \omega^{d_i}_{t+1} + \beta \sum \frac{n^{d_j}_{k'}}{n_{k'}} \omega^{dk'_j}_{t+1}$
    else
      $\omega_{t+1} \leftarrow \sum \frac{n_k}{n} \omega^k_{t+1}$
    end if
  end for

procedure WORKERUPDATE:(d, ω, k)
  $B \leftarrow$ (split data R^d_k into batches size of B)
  for each local epoch i from 1 to E **do**
    for batch $b \in B$ **do**
      $\omega \leftarrow \omega - \eta \cdot \nabla L(\omega)$
    end for
  end for
return ω to server

The proposed FL process involves four steps, as shown in Algorithm 1:

1) The broker server selects a set of workers $M_k = \{d_1, d_2, \cdots, d_{p_k}\}$ from the total available workers $\mathcal{D}_k$ for training, where p_k is the total number of workers selected.
2) The broker server starts a model training process with a preset of model parameter weights ω. Each worker d downloads the primary model with the weights and hyperparameters from the broker server. The hyperparamters include the local min-batch size B, the number of local epochs E, and the learning rate α.
3) The workers train the model locally using their own

data.

a) The workers split their data into batches of size B
b) During each local round of training, the workers utilize stochastic gradient descent $\nabla L(\omega)$ to calculate the loss and to compute new local weights.
c) After a set of local epochs E, each worker uploads the generated weights to the broker.

4) The broker gathers the locally trained models and aggregates them to obtain a shared global model.
   a) The broker utilizes the FedAvg method to aggregate worker weights by $\omega_{t+1} = \sum(n_d/n)\omega_t^d$ where ω_{t+1} is the global model weights, ω_{t+1}^d is worker d's model weights, n_d is worker d's local data size, and n is the global data size.
   b) The broker iterates step (2) until a preset epoch number E_k or an accuracy condition is reached.

The FedAvg method works under the assumption that the global data distribution on each cell is balanced, even though the local data on the workers may be disproportionate [5]. However, for crowdsourced COVID-19 symptom data, the global data distribution can be biased. For instance, there might be not enough data in some areas or the data disproportionately represent a certain group in a specific area. To overcome this issue, we propose a data pre-processing method and an privacy preserving adaptive worker selection algorithm, as discussed below.

B. Data Pre-processing

The workers prepare their datasets before the broker launches the training task. The data pre-processing method includes upsampling and regression imputation. If the workers have relatively small amounts of user data, the workers augment their data sizes by upsampling. If their datasets contain missing values, the workers apply a regression imputation, which replaces missing values with a predicted value based on a regression line.

C. Privacy Preserving Adaptive Worker Selection

We assume that each grid cell has an established age distribution P_k. A biased distribution occurs when the training data doesn't follow P_k. We observe that there normally exists at least another cell next to the cell g_k, which has the same or similar COVID-19 vulnerability level as cell g_k. Based on this observation, we introduce an adaptive worker selection algorithm that allows the broker b_k to find additional workers from cell g_k's neighboring workers to make the global distribution of collected data close to P_k. We use $g_{k'}$ to denote the cell g_k's neighbor cells. Often times there can be more than one neighboring cell.

Before selecting the new workers, the broker b_k first queries the age distribution $\widetilde{P}_k$ from the collected data of the selected workers. Moreover, to protect the sensitive age information of each participant, based on the SMC scheme (e.g., homomorphic encryption), each selected worker owns a private key sk_d to encrypt the DP version of age distribution information and the broker uses the key sk_0 to decrypt the statistics when receiving all of the cipher contents from the workers. As a random variable with Laplace distribution can be simulated using other random variable from the same distribution [10], the worker d_{p_m} can generate a random noise $z_k^m \sim \text{Lap}(\alpha)$ locally. Denote $\text{Lap}(\alpha) = -\alpha\ \text{sgn}(U)\ \ln(1-2|U|)$, where α is set to $\epsilon/\Delta f$, sgn denotes the sign function and U is a random variable with uniform distribution ranging $(-1/2, 1/2)$. In the adaptive worker selection scheme, since the broker b_k is interested in age distribution, the sensitivity Δf is $1/y$ and y is the dimension of the probability space. After generating the DP noise z_k^m, each worker sends the noisy data $\hat{P}_k^m = P_k^m + z_k^m$ encrypted with private key pk_d to the broker b_k. Then the broker decrypts the summation data $\hat{P}_k$ with the key sk_0 and the aggregated noise follows $\text{Lap}(\alpha) = \sqrt{B_{p_k} - 1}\sum \text{Lap}(\alpha)$ where the random variable B_{p_k} is taken from beta distribution with parameters 1 and $(p_k - 1)$ [10]. Thus, the broker only learns the summation of the age distribution under DP protection and no additional personal information can be inferred.

After receiving the age information $\hat{P}_k$ from the selected workers, it uses a greedy strategy to find more suitable workers. The broker searches the cell's neighboring workers to select the ones that make the combined distribution following the established age distribution P_k. It uses the KullBack-Leibler divergence (KLD) to measure the difference between the probability distributions of P_k and $P_b = (\hat{P}_k + P_{k'}^d)$, as follows:

$$\arg\min_i D_{KL}(P_k \| (\hat{P}_k + P_{k'}^d)), i \in D_{k'}, \tag{2}$$

where $P_{k'}^d$ is the worker's probability distribution from the neighboring cell $g_{k'}$. Once the broker finds a new worker, it updates its distribution P_b. It repeats this process until P_b is close to P_k or when a pre-defined number of total neighboring workers is met. Using adaptive worker selection, the broker b_k gathers the trained models from the workers inside and also outside of cell g_k. It ensembles the trained parameters from both types of workers, as shown below:

$$\omega_{t+1} \leftarrow \alpha \sum_{i=1}^{P_k} \frac{n_k^{d_i}}{n_k} \omega_{t+1}^{d_i} + \beta \sum_{j=1}^{P_{k'}} \frac{n_{k'}^{d_j}}{n_{k'}} \omega_{t+1}^{d_j}, \tag{3}$$

where $\alpha + \beta = 1$ and $\alpha \geq \beta$. The broker assigns more weights to the training results from its own cell g_k.

With the additional selected training data from neighborhood workers, a broker aggregates more training models, which will make the global model more reliable. It can also help mitigate the overfitting problem caused by small datasets in certain areas. The ensemble results improve the prediction accuracy and reduce error in the learning models.

We utilize the LSTM model to predict the COVID-19 vulnerability level. The LSTM model [7] is an extension of the recurrent neural network (RNN) model that introduces memory cells. Our model uses the memory cells to store COVID-19 symptoms for long periods of time. More accurate predictions can be achieved with the LSTM model as the

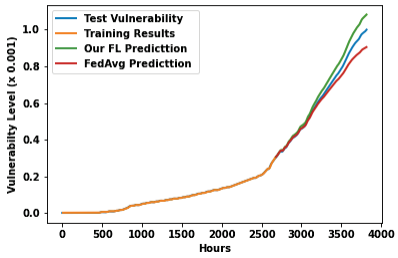

Fig. 2. Prediction accuracy.

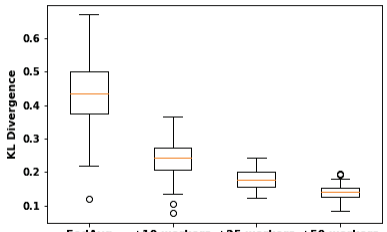

Fig. 3. KL divergence: $\widetilde{P}_k \| P_k$.

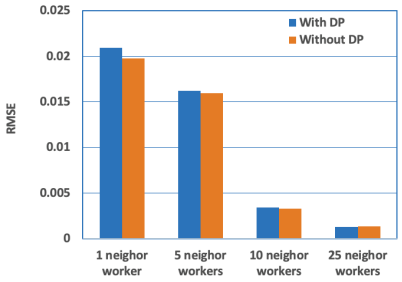

Fig. 4. RMSE with DP on $\widetilde{P}_k$.

prediction takes into account the COVID-19 symptom history data [11].

V. Performance Evaluation

We now examine the performance of the proposed framework for the construction of the privacy-preserving vulnerability map. The software used for performance evaluation is Python. We regard the City of Houston as the target area for estimating the risk level at each super neighborhood. Houston's government has divided the city into 88 super neighborhoods [12]. Each neighborhood has the attributes of a grid ID and boundary GPS coordinates. Each neighborhood's age distribution is utilized in the adaptive worker selection algorithm. The simulation results are based on the publicly available Demystata COVID-19 dataset [13] that includes COVID-19 cases within the US aggregated by zip code. We estimate the super neighborhood COVID-19 cases based on the percentage of population in the zip code. We simulate the user symptom data by using the number of COVID-19 cases in each age group reported by Harris County Public Health [14]. We develop a mobile crowdsourcing app that allows users to report symptoms and the app displays the COVID-19 vulnerability map. The mobile app is implemented using Android Studio and Google Maps.

Experiment setup The data bias scenario is an important aspect of federated learning. To evaluate our proposed approach to handle this problem, we simulate biased user symptom data in the experiment. We first generate a large number of users and then partition them equally into four age groups, which are the same age groups provided by the Houston super neighborhood report [12]. The groups are: "under 5 yrs", "5 to 17 yrs", "18 to 64 yrs" and "65 & over".

To simulate the balanced user age in each area, we sample users from each age group by following the age group distribution in each super neighborhood [12], and assign them into the corresponding area. To simulate the unbalanced user age data in an area, we assign a large number of users from a specific age group into the area, to change the area's age distribution. After that, we add Covid-like symptoms to each user based on the number of COVID-19 cases in each age group provided by Harris County Public Health [14]. To simulate the missing values in user data, we randomly select users and drop their data points at arbitrary timestamps. We finally assign the user data into the workers in each area.

TABLE I
Training Parameters

Notation	Description
M	total number of workers in a zip code
M_k	the number of workers selected in the training process
$M_{k'}$	the number of neighboring workers selected in the training process
E	local epoch
E_b	broker epoch

We use the simulated test set to conduct performance analysis. We allocate 70% of the data for training and the remaining 30% of the data for testing. Our scheme is evaluated by comparing it with the FedAvg FL algorithm as the baseline. We use the model training results for one area to demonstrate our proposed approach. Table 1 shows the notations of the parameters used in the experiment.

Prediction accuracy Figure 2 shows the prediction accuracy improvement in one area. The blue line is the vulnerability level from the testing data and the green line is our proposed approach and the red line represents the FedAvg results. Our results are much closer to the testing results with the root mean square error (RMSE) reduced from 0.372×10^4 to 0.125×10^4. On average there is 6% improvement in predication accuracy. FedAvg, a standard FL method, becomes less accurate when it uses underrepresented or imbalanced data to train ML model. This results in misleading conclusions and the model can not be trusted to provide a reliable prediction. Our approach provides a reliable prediction by using both the data pre-processing method and the adaptive worker selection algorithm to mitigate the biased dataset issue. The data pre-processing method includes upsampling of small datasets and regression imputation to estimate the missing values in the workers' user data.

Adaptive worker selection We use the distribution of $D_{KL}(P_k \| P_b)$ to present the changes of equilibrium degree when selecting more workers, as illustrated in Fig. 3. P_k is the given age distribution in area g_k and P_b is a broker's age distribution, which is obtained by the selected workers $(M_k + M_{k'})$ from both area g_k and the neighboring areas $g_{k'}$. The distribution results on the left are generated from the FedAvg approach, which uses only the local area workers;

while the results on the right three boxes use our adaptive worker selection approach to add more neighboring workers. The results show that the proposed algorithm can significantly re-balance the user age distribution, e.g., from 0.44 to 0.13, with an increase in neighboring workers. The results suggest that if balanced user data is selected from the workers, the broker can create a good global model with reduced bias to achieve accuracy improvement.

We also study the impact of applying differential privacy on the broker's $\widetilde{P}_k$ value with a smaller privacy budget $\epsilon = 0.6$, as shown in Fig. 4. We found that with the increase of neighboring workers (e.g. ≥ 10 workers), applying DP to $\widetilde{P}_k$ preserves the user age distribution information without deteriorating the final accuracy because the user age distribution becomes balanced with more workers.

Time overhead Our FL framework requires three major additional tasks: data pre-processing, adaptive worker selection and extra training epochs using neighboring workers. The time required for data pre-processing is negligible since it is conducted in the initialization phase before the training starts. The adaptive worker selection uses a greedy strategy. The time complexity of the searching process is $\mathcal{O}((d_{k'}^{M_{k'}})^2)$. A broker performs the adaptive worker selection only once when the user age distributions on workers are not dynamically and rapidly changing. Otherwise, the broker needs to search for neighboring workers in each E_b training round. In FL learning, the most time-consuming part is training a local model in each worker. We use T to denote the average training time of a local epoch in a worker. The total time spent on the training round is $E_b \times E \times T$.

Vulnerability prediction map Figure 5 and Figure 6 show our mobile app that displays the Google map with the estimated average vulnerability of COVID-19 in the 88 super neighborhoods of Houston. Fig. 5 utilizes the collected user symptom data to directly estimate the vulnerability risk level. Fig. 6 depicts a week-long future vulnerability trend predicted by the LSTM model. The future trend reflects the estimated percentage of vulnerable population towards COVID-19.

VI. Conclusion

We have developed a novel FL framework that constructs a privacy preserving fine-grained COVID-19 vulnerability map by leveraging multiple crowdsourcing mobile and web apps. The FL-based fine-grained vulnerability map construction utilizes a large amount of survey data available while at the same time providing a privacy guarantee. It has provided a more reliable vulnerability prediction than existing methods. The potential imbalanced or biased datasets from each individual self-reporting app has been addressed by an adaptive worker selection algorithm that ensures the aggregated age distribution correctly represents the age distribution in the fine-gained area. A dataset pre-processing mechanism has been also employed to mitigate the potential small dataset problem. To protect user's privacy, a DDP scheme has been applied to the age distribution parameter. The simulation results of the proposed FL framework with LSTM training models show a 6% improvement in accuracy over the widely used FedAvg algorithm.

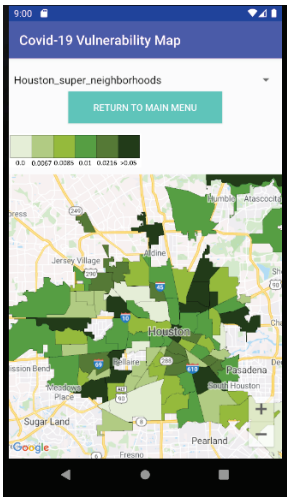

Fig. 5. Vulnerability map.

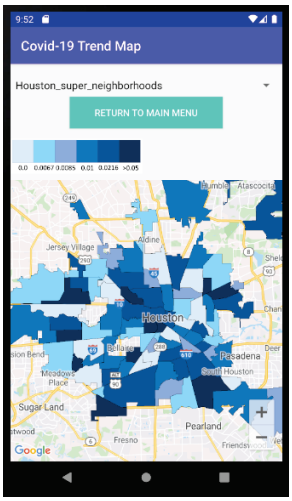

Fig. 6. Future trend map.

References

[1] R. Chen, L. Li, J. Chen, R. Hou, Y. Gong, Y. Guo, and M. Pan, "Covid-19 vulnerability map construction via location privacy preserving mobile crowdsourcing," in *IEEE Global Communications Conference(GLOBECOM'20)*, Taipei, Taiwan, December 2020.

[2] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, "Self-balancing federated learning with global imbalanced data in mobile systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 59–71, 2020.

[3] "Facebook: COVID-19 interactive map & dashboard," https://covid-survey.dataforgood.fb.com/survey_and_map_data.html, Accessed October, 2020.

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

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics(AISTATS'17)*, Ft. Lauderdale, FL, April 2017, pp. 1273–1282.

[6] B. Ikiz, "A pandemic ai engine without borders," https://hai.stanford.edu/blog/pandemic-ai-engine-without-borders, Accessed August, 2020.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, Xi'an, China, April 2008.

[9] R. Chen, X. Zhang, J. Wang, Q. Cui, W. Xu, and M. Pan, "Data-driven small cell planning for traffic offloading with users' differential privacy," in *IEEE International Conference on Communications (ICC'20)*, Dublin, Ireland, June 2020, pp. 1–6.

[10] S. Goryczka, L. Xiong, and V. Sunderam, "Secure multiparty aggregation with differential privacy: a comparative study," in *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, Genoa, Italy, March 2013, pp. 155–163.

[11] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons & Fractals*, 2020.

[12] C. of Houston, "Super Neighborhood," https://www.houstontx.gov/superneighborhoods, Accessed May, 2020.

[13] "Demystdata: Covid-19," https://www.snowflake.com/datasets/demystdata-covid-19, Accessed October, 2020.

[14] "Harris county houston COVID-19 cases," https://publichealth.harriscountytx.gov/Resources/2019-Novel-Coronavirus, Accessed October, 2020.