Privacy-Preserving Collaborative Learning for Mobile Health Monitoring

Yanmin Gong, Yuguang Fang
Department of Electrical and Computer Engineering
University of Florida
Gainesville, Florida 32611 USA
Email: {ymgong@, fang@ece.}ufl.edu

Yuanxiong Guo
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK, 74078 USA
Email: richard.guo@okstate.edu

***Abstract*—Health monitoring is an important category of mobile Health (mHealth) applications. Users generate a large volume of data during health monitoring, which can then be used by the mHealth server for constructing diagnosis or prognosis prediction models. However, these training samples contain private information of data owners, who may be reluctant to share them with the mHealth server. This paper proposes and experimentally studies a scheme that keeps the training samples private while enabling accurate construction of diagnosis and prognosis models. We specifically consider logistic regression models which are widely used in mHealth, and decompose the logistic regression model construction problem into small subproblems that can be executed by each user using their own private data. In this manner, users can keep their raw data locally and only upload encrypted parameters to the mHealth server for model construction. We show that our scheme suits well in mHealth applications by conducting experimental evaluations based on a real-world dataset and analyzing its computation overhead.**

I. Introduction

Mobile health (mHealth) technologies, including remote monitoring, wearable devices, and embedded sensors, have grown rapidly in the past years and shown great potential to improve the quality and efficiency of healthcare. In mHealth, long-term and continuous health monitoring is enabled by mobile devices that wirelessly connect body sensors, producing a high volume of physiological or physical activity data, such as electrocardiogram, glucose concentration, breathing rate, and body motion. These data are sent from mobile devices to the mHealth server, who runs monitoring programs for patients with chronic conditions, individuals desiring to change behavior, or doctors detecting physiological or behavioral aberrations for early diagnosis. Health monitoring enalbes timely intervention and better management of individual health status, thus significantly improving healthcare quality. In this paper, we use the term "Patient" to represent the subject of sensing in all the previous scenarios, where the first character of the term is capitalized to remind readers of its broad meaning.

The health monitoring data contain valuable information that can not only be used for querying health monitoring programs, but also be used for constructing disease diagnosis or prognosis models. In this paper, we particularly consider the logistic regression, a classic machine learning technique that is commonly used in medical diagnosis and prognosis, particularly for modeling disease state (healthy or unhealthy) and decision making (yes or no) [1]. Due to the diversity of human physiology, classifiers trained on a single-user dataset may not be robust to a wide range of input data. Collaborative learning [2] overcomes this limitation by utilizing multiple-user datasets which contain enough diversity. In collaborative learning, multiple individuals confide their data to a centralized party (e.g., a cloud server or a research institution) as training samples. For disease diagnosis or prognosis models, the data may be sensing data from patients with the same disease. The centralized party could then construct models based on the sensing data.

Patients who participate in collaborative learning should send their health monitoring data to a centralized party (here we call it as the mHealth server). However, sharing these data raises severe privacy concerns. First, a wide range of sensing data, both physiological and physical, are collected through health monitoring. The physiological data reflect the health status of Patients and the physical activity data may reveal sensitive information about their lifestyles and activities, both raising privacy concerns. Second, data collection process in health monitoring is usually continuous and long-term. The resulting data would have much higher volume than the medical data collected through visiting doctors. Third, the mHealth applications could be run by a wide range of parties including doctors, insurance companies, diet advisers, athletic coaches or home-care providers. In such setting, Patients may not trust the mHealth server in sharing their private data. Hence, to incentivize users to contribute their data for model construction, we should provide privacy guarantees.

Privacy-preserving learning has attracted a lot of attentions in literature. One of the most popular way is to anonymize the data by hiding the identity of the data source [3]. However, it is possible to re-identify the data source. Another approach is to perturb the data content before transmitting it to the centralized party [4]. However, perturbation always introduces error in the modeling process, trading accuracy for privacy. Secure multi-party computation-based approach is a conventual approach for training classifiers based on multi-party collection of private data [5]. However, the cryptographic techniques used in secure multi-party computation usually incur high computation

This work was supported in part by the U.S. National Science Foundation under grants CNS-1423165.

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

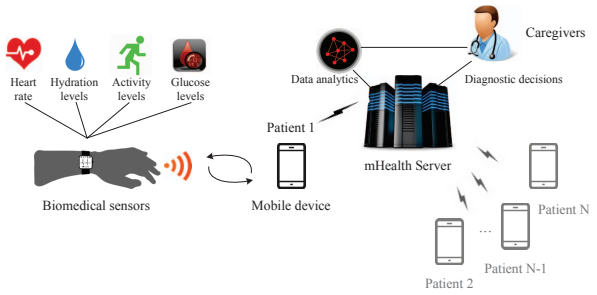


Fig. 1. Architecture of Mobile Health Monitoring.

cost, which is unpractical for mobile health applications. Gentry [6] proposes fully homomorphic encryption for privacy-preserving computation, however, current solutions for fully homomorphic encryption are not quite efficient [7].

In this paper, we develop a privacy-preserving collaborative learning scheme that utilizes health monitoring data from multiple Patients towards training disease diagnosis and prognosis models. Specifically, we consider logistic regression for model training. We leverage the intrinsic structure of the logistic regression model and decompose the collaborative learning problem into multiple subproblems that can be solved independently using local data. An aggregate classifier is then computed by averaging local trained parameters. The local training and averaging steps are repeated multiple rounds until the aggregate classifier converges. This iteration process enables the aggregate classifier to adapt to new sensing data, which is especially suitable to handle continuous data streams in mHealth monitoring. Our scheme is highly efficient and incurs low computation overhead for each Patient, thus scalable to a large number of Patients.

The remainder of this paper is organized as follows. We first present the system model in Section II. Then we develop a scheme to privately construct the logistic regression model in Section III. Section IV presents experimental results and performance analysis. Section V concludes the work.

II. System Model

In this section, we first outline the system architecture for mHealth monitoring and describe the threat model and design goals. We then give some background on logistic regression and present a motivating scenario in which private computation on Patient data is desirable.

A. System Architecture

We focus on the patient-centered mHealth systems where Patients share their private sensing data to an mHealth server for model training. The system model is shown in Fig. 1. As shown in the figure, the Patient is continuously monitored by multiple sensors, generating a large volume of data such as heart rates, hydration levels, activity levels, and glucose levels. The sensors are wirelessly connected to a mobile device, which collects and stores the sensed data. The raw sensing data in a certain time period may be preprocessed and transformed into a feature vector. Each feature vector is associated with a binary outcome variable. Specifically, each Patient i poses a dataset $D_i := \{(a_{ij}, b_{ij}), j = 1, \ldots, m_i\}$, where $a_{ij} \in \mathbb{R}^n$ is a feature vector, $b_{ij} \in \{-1, 1\}$ is the corresponding label of the outcome variable, and m_i is the number of instances owned by user i. The mHealth server performs data analysis based on feature vectors from a variety of Patients and train health monitoring programs. The health monitoring programs can assist caregivers in making diagnostic decisions based on feature vectors generated from real-time biomedical sensing data. The health monitoring program we discuss in this paper is a logistic regression model which enables computation of the outcome's probability given the feature vector. The model is collaboratively learned by datasets from multiple users.

A motivating scenario would be diabetes management. Suppose a research institute wants to construct a predictive model that predicts whether the glucose level would be normal or not given certain risk factors. The institute recruits a group of Patients for their study. As part of the study, a Patient wears a mobile device provided by the institute that continuously monitors factors including medication, physical activity, food intake, and other biological and environmental factors. The patient also records his blood glucose levels at fixed frequencies (e.g., three times per day) and labels the blood glucose levels as either 'positive" or "negative" by comparing them to a safety threshold. Both user data and labels are sent to the institute, who trains a model that predicts whether the blood glucose level is above or below the safety threshold. Such model would help diabetics to better monitor their blood glucose and reduce the frequencies of unpleasant blood tests.

B. Threat Model and Design Goals

During collaboratively model training, each Patient contributes a set of training samples. The training samples are considered private as they may reveal sensitive information such as health status and unusual activities of individuals. Each training sample consists of a feature vector and a label, and we consider both as private. We assume that an attacker wants to learn the private information contained in training samples. The attacker could be either the mHealth server or an outsider who compromises the mHealth server. Meanwhile, we assume that correctly computing the model is for the best interest of the mHealth server, and thus it would not try to distort the result of the training process. Moreover, the resulting model is public and shared by the mHealth server and all Patients. We also assume secure communication between body sensors and mobile devices. Such secure communication channel can be provided through encryption such as the one in [8], which is out of the scope of our paper.

There are several challenges in designing a privacy-preserving scheme for collaborative learning. First, since mHealth server is not trusted to learn the training samples, it is best to keep the training samples locally at each Patient. In this case, *how can we design a collaborative learning scheme based on distributed data without sacrificing learning accuracy?* We will resolve this issue with a distributed algorithm which iteratively trains local classifiers and constructs aggregate classifiers based on local classifiers. Second, even if data can be locally trained, the resulting local classifiers still

need to be aggregated at the mHealth server in each iteration. These local classifiers are trained based on private personal data and reveal sensitive information about the Patient. Then, *how can we ensure no private information is leaked during the aggregation process?*

Besides the challenges in achieving privacy described above, the scheme should also be scalable to a large number of users to keep significant diversity in training samples. The main factor that influences scalability is the computational complexity. Hence, we want to keep the computation overhead at the Patient side low even with a large number of participating Patients. Moreover, our scheme should be efficient for samples generated over multiple time periods. The learning model should be periodic updated when new training samples arrive. This observation motivates us to design a scheme with low amortized computational overhead (i.e., the average computation cost for each time period).

C. Logistic Regression

Logistic regression is a classic machine learning technique that is commonly used in medical diagnosis and prognosis. Here we briefly discuss the basics of logistic regression.

Given a set of labeled training samples $\cup_{i=1}^{N} D_i$, the ℓ_1 regularized logistic regression problem [9] is defined as

$$\min \sum_{i=1}^{N} \sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(a_{ij}^T\mathbf{w}+v\right)\right)\right)+\lambda\left\|\mathbf{w}\right\|_1, \quad (1)$$

with two optimization variables: the weight vector $\mathbf{w} \in \mathbb{R}^n$ and the intercept $v \in \mathbb{R}$. Here $\lambda > 0$ is the regularization parameter.

With the trained regularized logistic regression classifier $(\mathbf{w}, v)$, logistic regression models the probability distribution of the class label $y \in \{-1, 1\}$ given a feature vector $\mathbf{x} \in \mathbb{R}^n$ as follows:

$$\Pr(y=1|\mathbf{x};\mathbf{w},v)=\frac{1}{1+\exp\left(-\left(\mathbf{x}^T\mathbf{w}+v\right)\right)}. \quad (2)$$

$$\Pr(y=-1|\mathbf{x};\mathbf{w},v)=\frac{\exp\left(-\left(\mathbf{x}^T\mathbf{w}+v\right)\right)}{1+\exp\left(-\left(\mathbf{x}^T\mathbf{w}+v\right)\right)}. \quad (3)$$

The resulting classifier can predict the class label of new feature vectors, which is particularly suitable for disease state prediction (healthy or unhealthy) and decision making (yes or no), and thus is commonly used in medical diagnosis and prognosis.

In [10], Tabaei and Herman conducts a diabetes study which screens diabetes based on logistic regression classifiers. In their study, each Patient generates a private feature vector, which consists of age (years), sex (0 = male and 1 = female), body mass index (BMI), postprandial time (PT), random capillary plasma glucose level (RPG). Each feature vector is associated with a label b_{ij}, which is an indicator of fast plasma glucose (FPG) and plasma glucose 2h after a 75g oral glucose load (2-h PG), both indicating the risk of having diabetes. Specifically, $b_{ij} = 1$ when FPG $\geq$ 140 mg/dl or 2-h PG $\geq$ 200 and $b_{ij} = -1$ otherwise. The mHealth server collects data from $1,032$ Patients and use the data to train a logistic regression classifier. The resulting classifier is $(\mathbf{w}, v)$, where $\mathbf{w} = [0.0331, 0.0308, 0.2500, 0.5620, 0.0346]$ and $v = -10.0382$. Given a feature vector x, the classifier can predict the probability that FPG $\geq$ 140mg/dl or 2-h PG $\geq$ 200 according to (2) and (3).

III. Privacy-Preserving Model Training Via Collaborative Learning

In this section, we describe a scalable and practical distributed scheme that enables collaborative training of logistic regression models. We first describe alternating direction method of multipliers (ADMM), which is the basis of our scheme.

A. Basics of ADMM

Alternating direction method of multipliers (ADMM) is a distributed algorithm that solves a large-scale optimization problem by decomposing it into smaller subproblems that are easier to solve [11]. The algorithm solves problems in the following form:

$$\begin{aligned} \underset{x,z}{\text{minimize}} \quad & f(x)+g(z) \\ \text{subject to} \quad & Ax+Bz=c \\ & x\in\mathcal{X}, z\in\mathcal{Z} \end{aligned} \quad (4)$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, and $c \in \mathbb{R}^p$. We assume that functions f and g are convex, and $\mathcal{X}$ and $\mathcal{Z}$ are non-empty polyhedral sets. The variables are split into two parts x and z, and the objective function is separable across the splitting.

We can form the augmented Lagrangian for (4) as

$$\begin{aligned} L_\rho(x,z,y) =& f(x)+g(z)+y^T(Ax+Bz-c) \\ &+(\rho/2)\left\|Ax+Bz-c\right\|_2^2, \end{aligned} \quad (5)$$

where $\rho > 0$ is the penalty parameter and the last term is the regularization term. We can view the augmented Lagrangian as the Lagrangian associated with the following problem

$$\begin{aligned} \underset{x,z}{\text{minimize}} \quad & f(x)+g(z)+(\rho/2)\left\|Ax+Bz-c\right\|_2^2 \\ \text{subject to} \quad & Ax+Bz=c, \\ & x\in\mathcal{X}, z\in\mathcal{Z}. \end{aligned} \quad (6)$$

Since the regularization term equals zero for any feasible x and z, the above problem is equivalent to problem (4). The introduced regularization term ensures that L is strictly convex even when f and g are affine and helps to improve the convergence property of the algorithm.

ADMM consists of three steps in each iteration k:

1) x-minimization with z and y fixed:

$$x^{k+1} := \underset{x\in\mathcal{X}}{\operatorname{argmin}} L_\rho(x, z^k, y^k). \quad (7)$$

2) z-minimization with x and y fixed:

$$z^{k+1} := \underset{z\in\mathcal{Z}}{\operatorname{argmin}} L_\rho(x^{k+1}, z, y^k). \quad (8)$$

3) Dual variable y update:

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \quad (9)$$

where the step size equals to the penalty parameter ρ.

Note that in ADMM, x and z are updated sequentially instead of jointly as in dual ascent algorithm. The order of x-update step and z-update step can be reversed, leading to a variation on ADMM. The optimality and convergence of the ADMM algorithm is given by the following proposition, and its proof can be found in [12].

***Proposition* 1.** *Assume that the optimal solution set of* (4) *is non-empty, and either $\mathcal{X}$ is bounded or A^TA is nonsingular. Then a sequence $\{x^k, z^k, y^k\}$ generated by the iterations* (7)(8)(9) *is bounded, and every limit point of $\{x^k, z^k\}$ is an optimal solution of* (4).

In practice, ADMM usually converges to modest accuracy within a few tens of iterations.

B. Our Algorithm

The problem (1) cannot be solved directly by ADMM since the objective function is not separable over two sets of variables. To address this challenge, we introduce a set of auxiliary variables $(\mathbf{w}_i, v_i), \forall i$ and reformulate the optimization problem as

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(a_{ij}^T\mathbf{w}_i + v_i\right)\right)\right) + \lambda\left\|\mathbf{w}\right\|_1 \\ \text{s.t.} \quad & \mathbf{w}_i = \mathbf{w}, v_i = v, i \in \mathcal{N} \end{aligned} \tag{10}$$

It is obvious that the new problem (10) is equivalent to the original problem (1). Note that the objective function in the problem (10) is now separable over two sets of variables $(\mathbf{w}_i, v_i), \forall i$ and $(\mathbf{w}, v)$. We can view $(\mathbf{w}_i, v_i)$ as the copy of regression parameters at each Patient $i \in \mathcal{N}$, and $(\mathbf{w}, v)$ as the copy of regression parameters at the mHealth server side. These two sets of variables are connected through equality constraints.

In the following, we demonstrate that through these auxiliary variables the problem can be decomposed. For simplicity of notation, we define $\alpha := \{(\mathbf{w}, v)\}$ and $\beta := \{(\mathbf{w}_i, v_i), \forall i \in \mathcal{N}\}$. The augmented Lagrangian of (10) is

$$\begin{aligned} L_\rho(\alpha,\beta,\gamma) = & \sum_{i=1}^{N}\sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(a_{ij}^T\mathbf{w}_i + v_i\right)\right)\right) \\ & + \lambda\left\|\mathbf{w}\right\|_1 + \sum_{i=1}^{N}\left((\mathbf{w}_i-\mathbf{w})^T\boldsymbol{\gamma_{i,w}} + \gamma_{i,v}(v_i - v)\right) \\ & + \sum_{i=1}^{N}(\rho/2)\left((\mathbf{w}_i-\mathbf{w})^T(\mathbf{w}_i-\mathbf{w}) + (v_i-v)^2\right), \end{aligned}$$

where $\gamma := \{(\boldsymbol{\gamma_{i,w}}, \gamma_{i,v}), \forall i \in \mathcal{N}\}$ are the dual variables corresponding to the constraints in (10).

We can then solve the problem by updating α, β, and γ sequentially. Specifically, at the $(k+1)$-th iteration, the α-minimization step involves solving the following problem:

$$\begin{aligned} \min_{\alpha} \quad & \lambda\left\|\mathbf{w}\right\|_1 + (\rho N/2)\mathbf{w}^T\left(\mathbf{w} - 2\overline{\mathbf{w}}^k - 2\overline{\boldsymbol{\gamma}}_w^k/\rho\right) \\ & + (\rho N/2)v\left(v - 2\overline{v}^k - 2\overline{\gamma}_v^k/\rho\right), \end{aligned} \tag{11}$$

where the overline notation denotes the average of a vector over $i = 1, \ldots, N$. A closed-form solution of the above problem can be computed using subdifferential calculus [13]. Specifically, the optimal solution is given by

$$\begin{aligned} \mathbf{w}^{k+1} := & \left[\overline{\mathbf{w}}^k + \overline{\boldsymbol{\gamma}}_w^k/\rho - (\lambda/\rho N)\right]_+ \\ & - \left[-\overline{\mathbf{w}}^k - \overline{\gamma}_w^k/\rho - (\lambda/\rho N)\right]_+ \end{aligned} \tag{12a}$$

$$v^{k+1} := \overline{v}^k + \overline{\gamma}_v^k/\rho, \tag{12b}$$

where the operator $[\cdot]_+$ means taking the maximum of zero and the argument inside.

After obtaining α^{k+1} from the α-minimization step, the β-minimization step consists of solving the following:

$$\begin{aligned} \min_{\beta} \quad & \sum_{i=1}^{N}\sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(a_{ij}^T\mathbf{w}_i + v_i\right)\right)\right) \\ & + \sum_{i=1}^{N}(\rho/2)\mathbf{w}_i^T(\mathbf{w}_i - 2\mathbf{w}^{k+1} + 2\boldsymbol{\gamma_{i,w}}^k/\rho) \\ & + \sum_{i=1}^{N}(\rho/2)v_i(v_i - 2v^{k+1} + 2\gamma_{i,v}^k/\rho), \end{aligned} \tag{13}$$

which is decomposable over all Patients. Effectively, each patient i only needs to independently solve the following subproblem:

$$\begin{aligned} \min_{\beta_i} \quad & \sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(a_{ij}^T\mathbf{w}_i + v_i\right)\right)\right) \\ & + (\rho/2)\mathbf{w}_i^T(\mathbf{w}_i - 2\mathbf{w}^{k+1} + 2\boldsymbol{\gamma_{i,w}}^k/\rho) \\ & + (\rho/2)v_i(v_i - 2v^{k+1} + 2\gamma_{i,v}^k/\rho). \end{aligned} \tag{14}$$

This per-Patient subproblem has a much smaller scale and uses the Patient's own private information. Standard methods such as Newton's method or the conjugate gradient method can be applied to solve the subproblem efficiently.

Having obtained α^{k+1} and β^{k+1}, the dual update is as follows:

$$\boldsymbol{\gamma_{i,w}}^{k+1} := \boldsymbol{\gamma_{i,w}}^k + \rho\left(\mathbf{w}_i^{k+1} - \mathbf{w}^{k+1}\right) \tag{15a}$$

$$\gamma_{i,v}^{k+1} := \gamma_{i,v}^k + \rho\left(v_i^{k+1} - v^{k+1}\right). \tag{15b}$$

The entire procedures of our algorithm are described in Algorithm 1. Obviously, our problem meets the conditions in Proposition 1, and the proposed algorithm converges to the optimal solution. At the end of the algorithm, each user i will learn the global optimal classifiers $\mathbf{w}$ and v without sending his local dataset to others. Therefore, this system can preserve user privacy without sacrificing the utility of the learning function.

C. Private Aggregation of Local Regression Parameters

In line 4 of Algorithm 1, the mHealth server needs to know the local regression parameters of all Patients. However, these local regression parameters are trained on individual private data and may leak sensitive information such as some statistics of the individual. We observe that in each iteration, the server only needs to know the average of these local regression parameters. Therefore, we use an aggregator-oblivious

Algorithm 1 Distributed Algorithm for Training Logistic Regression Model

1: The mHealth server initializes $k \leftarrow 0$, $\overline{\mathbf{w}}^0 \leftarrow 0$, $\overline{v}^0 \leftarrow 0$.
2: Each Patient i initializes $k \leftarrow 0$, $\boldsymbol{\gamma_{i,w}}^0 \leftarrow 0$, and $\gamma_{i,v}^0 \leftarrow 0$.
3: **repeat**
4: The mHealth server gathers $(\mathbf{w}_i^k, v_i^k)$ and $(\boldsymbol{\gamma_{i,w}}^k, \gamma_{i,v}^k)$ from all Patients $i \in \mathcal{N}$, and averages them to get $\overline{\mathbf{w}}^k$, $\overline{v}^k$, $\overline{\boldsymbol{\gamma}}_w^k$, and $\overline{\gamma}_v^k$. Then it updates $\mathbf{w}^{k+1}$ and v^{k+1} according to (12), and broadcasts them to all Patients.
5: After receiving $\mathbf{w}^{k+1}$ and v^{k+1}, each Patient i solves the per-Patient subproblem (14) independently using his own training dataset, and then updates independently the dual variables according to (15).
6: Each Patient sends the optimal solution $(\mathbf{w}_i^{k+1}, v_i^{k+1})$ and $(\boldsymbol{\gamma_{i,w}}^k, \gamma_{i,v}^k)$ to the mHealth server.
7: $k \leftarrow k+1$
8: **until** Convergence criteria is met

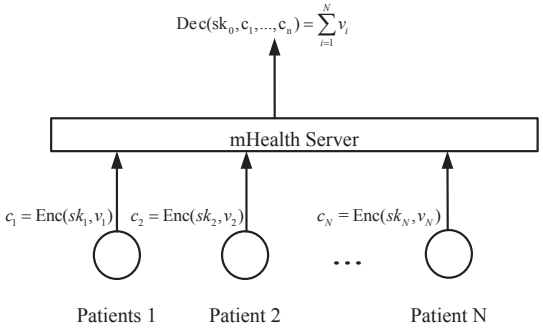


Fig. 2. Aggregation of Private User Data.

aggregation approach where the mHealth server can obtain the average value without knowing local regression parameters.

Specifically, the aggregation process iterates multiple times until convergence. In the k-th iteration, each user i has private values $\mathbf{w}_i^k$,v_i^k and α_i^{k-1}. The mHealth server wants to calculate the average values $\overline{\mathbf{w}}^k$, $\overline{v}^k$, and $\overline{\alpha}^{k-1}$. When context is clear, we omit the superscript k. We only describe how to compute the average of private value $\overline{v}$ from private data v_i since the aggregation process for the other two variables are the same. For ease of representation, we assume v_i is an integer. Aggregation of vectors can be treated as aggregating scalars at each component of the vectors. When v_i is a real number, a given precision is chosen in advance, and real numbers at the precision can be scaled by the corresponding factor to make them integers for encoding, as described in [14].

The individual private data are usually kept secret through encryption. For calculating average value of encrypted data, additive homomorphic encryption schemes (e.g., Paillier [15]) seems suitable, where private data of each user are encrypted with the public key of the mHealth server who can decrypts the aggregated ciphertexts. However, if the mHealth server gets the individual ciphertexts, it can directly decrypt individual ciphertexts and learn the private data.

Here we provide a modified homomorphic encryption scheme to protect the privacy of local regression parameters through aggregation. Our scheme borrows the idea of an aggregation scheme designed by Shi et al. [16]. The overview of our construction is shown in Fig. 2. At the beginning of the aggregation process, each Patient i has a secret sk_i. The patient encrypts his private data v_i with the secret sk_i and sends the ciphertext to the mHealth server. The mHealth server sums the the ciphertexts of v_i from all Patients, and obtains a ciphertext of $\sum_i v_i$ encrypted by its own secret sk_0. We summarize the scheme below.

Let $\mathbb{G}$ denote a cyclic group of prime order p for which Decisional Diffie-Hellman problem is hard. Let $H : \mathbb{Z} \rightarrow \mathbb{G}$ denote a hash function.

- *Key generation:* A trust authority chooses a random generator $g \in \mathbb{G}$ and random secrets $sk_1, \ldots, sk_N \in \mathbb{Z}_p$. The public parameter is g. Each user i obtains a private key sk_i , and the mHealth server obtains its private key $sk_0 = -(sk_1 + \ldots + sk_N)$.
- *Encryption:* During iteration k, user i encrypts its private value v_i as follows:

$$c_i \leftarrow g^{v_i} \cdot H(k)^{sk_i}.$$

- *Decryption:* Given the ciphertext $c_1, c_2, \ldots, c_n$, compute

$$P \leftarrow H(k)^{sk_0} \prod_{i=1}^{n} c_i,$$

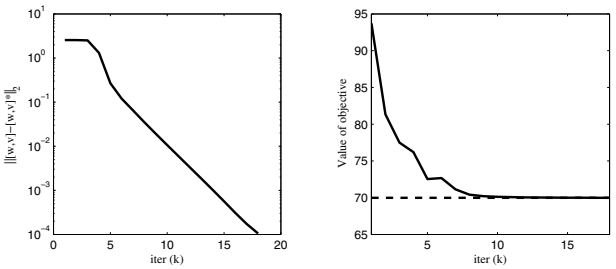
where $P = H(k)^{sk_0} \prod_{i=1}^n c_i = H(k)^{\sum_{i=0}^n sk_i} \cdot g^{\sum_{i=1}^n} = g^{\sum_{i=1}^n}$. The sum of v_i can then be calculated by computing the discrete log of P base g.

The scheme allows the untrusted mHealth server to periodically estimate the sum of v_i without knowing individual value of v_i. The average $\overline{v_i}$ can then be readily calculated by dividing the sum by the total number of users N. In each iteration, each Patient i computes $H(k)^{sk_i}$, and the mHealth server computes $H(k)^{sk_0}$. Since $\sum_i sk_i = 0$, we have $\prod_i H(k)^{sk_i} = 1$. In this way, the Patients do not need to communicate with each other to obtain their own secrets after the initial key generation process. Hence, the computation overhead of secret does not increase with the sensing time, achieving low amortized computation overhead.

The computation overhead for the aggregation process in each iteration comes the encryption and decryption process. Encryption operation in the construction includes one hash, one multiplication in a Diffie-Hellman group, and two modular exponentiations. The two modular exponentiations consumes much more time than the other operations, and thus dominates the running time. According to the benchmarking report of eBACs project [17], it takes around 0.3ms to compute a modular exponentiation using high-speed elliptic curves on a modern 64 bit computer. Hence, the construction is practical and poses low computation overhead for the Patients. The decryption process requires computation of a discrete log. Suppose we use Pollard's lambda method to solve this problem, the running time would be $\sqrt{N\Delta}$, where N is the number of users and Δ is the size of the plaintext space.

IV. Performance Evaluation

In this section, we evaluate the performance of our algorithm using a real-world dataset: the Pima Indians Diabetes



(a) classifier (b) objective function value

Fig. 3. Simulation results of our proposed distributed algorithm.

Dataset [18]. This dataset contains 768 records. Since some records have biologically impossible features such as zero-value blood pressure, we only select 336 records that do not contain zero-value features (111 belonging to class 1 and 225 belonging to class -1). Each record includes a 8-dimensional feature vector representing age, times of pregnancy, 2-h PG, diastolic blood pressure, triceps skin fold thickness, 2-h serum insulin, BMI, and diabetes pedigree function. Each feature vector has a corresponding label $+1$ (test positive for diabetes) or -1 (test negative for diabetes). To show the accuracy of our algorithm, we consider a centralized baseline algorithm that is privacy-oblivious. In the baseline algorithm, the mHealth server has access to all Patient data and solves the logistic regression problem directly. In our simulations, we implement our algorithm and the baseline algorithm in MATLAB and use CVX package [19] to solve convex optimization problems. The regularization parameter is set to be 1.

Fig. 3a shows the change of the logistic regression classifier with respect to the iteration number k. The y-axis of the plot represents the norm of the difference between regression parameters in each iteration and the global optimal regression parameters. The global optimal parameters are obtained by the baseline algorithm. We can see from the figure that the logistic regression classifier obtained by our algorithm converge fast to the global optimal classifier after 15 iterations. Fig. 3b shows the change of the objective value by iteration. The solid line indicates the objective value obtained by our algorithm, and the dashed line denotes the optimal objective value obtained by the baseline algorithm. As shown in the figure, the objective value of our algorithm decreases fast in the first few iterations and finally approaches the global minimum after 15 iterations. Note that the objective value does not necessarily decrease monotonically because at the first few iterations the logistic regression parameters may not satisfy the constraints in (10).

On a 64-bit notebook with 1.6GHz CPU, the computation time for all Patients is 0.13 seconds, which is very small. The baseline algorithm for logistic regression, on the other hand, uses 64.5 seconds on the same computer. Hence our algorithm is much faster than the baseline algorithm. Our algorithm converges fast to the global optimal solution, which results in small computation overhead for each Patient.

V. Conclusion

In this paper, we have proposed a private scheme for composing a logistic regression classifier based on distributed private mHealth data. Our scheme enables users to control their raw data and only share necessary features during the training process. We have further provided a solution to protecting the private information of local classifiers in the aggregation process. Experimental results on Pima Indians Diabetes data show that the proposed algorithm converges quickly and provides performance closely to the optimal result. Our scheme has low computation overhead for mobile devices, and thus is practical for mHealth monitoring scenarios. We have focused on the logistic regression problem in this paper. However, our scheme could also be generalized to other classification problems in mHealth applications, which constitutes our future work.

References

[1] S. C. Bagley, H. White, and B. A. Golomb, "Logistic regression in the medical literature:: Standards for use and reporting, with particular attention to one medical domain," *Journal of clinical epidemiology*, vol. 54, no. 10, pp. 979–985, 2001.

[2] K. A. Bruffee, *Collaborative learning: Higher education, interdependence, and the authority of knowledge*. ERIC, 1999.

[3] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*. IEEE, 2005, pp. 620–629.

[4] P. K. Fong and J. H. Weber-Jahnke, "Privacy preserving decision tree learning using unrealized data sets," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 2, pp. 353–364, 2012.

[5] J. Vaidya and C. Clifton, "Privacy-preserving decision trees over vertically partitioned data," in *Data and Applications Security XIX*. Springer, 2005, pp. 139–152.

[6] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[7] T. Graepel, K. Lauter, and M. Naehrig, "Ml confidential: Machine learning on encrypted data," in *Information Security and Cryptology–ICISC 2012*. Springer, 2013, pp. 1–21.

[8] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008, pp. 148–153.

[9] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd ed. Springer, 2009.

[10] B. P. Tabaei and W. H. Herman, "A multivariate logistic regression equation to screen for diabetes development and validation," *Diabetes Care*, vol. 25, no. 11, pp. 1999–2003, 2002.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[13] R. T. Rockafellar, *Convex analysis*. Princeton University Press, 1970.

[14] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data," *Journal of biomedical informatics*, vol. 50, pp. 234–243, 2014.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptologys*. Springer, 1999, pp. 223–238.

[16] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Network & Distributed System Security Symposium (NDSS)*, 2011.

[17] D. J. Bernstein and T. Lange, "ebacs: Ecrypt benchmarking of cryptographic systems," 2009.

[18] C. Black and C. Merz, "Pima indians diabetes," 1998. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

[19] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.