Private Data Analytics on Biomedical Sensing Data Via Distributed Computation

Yanmin Gong, *Student Member, IEEE,* Yuguang Fang, *Fellow, IEEE,* and Yuanxiong Guo, *Member, IEEE*

Abstract—Advances in biomedical sensors and mobile communication technologies have fostered the rapid growth of mobile health (mHealth) applications in the past years. Users generate a high volume of biomedical data during health monitoring, which can be used by the mHealth server for training predictive models for disease diagnosis and treatment. However, the biomedical sensing data raise serious privacy concerns, because they reveal sensitive information such as health status and lifestyles of the sensed subjects. This paper proposes and experimentally studies a scheme that keeps the training samples private while enabling accurate construction of predictive models. We specifically consider logistic regression models which are widely used for predicting dichotomous outcomes in healthcare, and decompose the logistic regression problem into small subproblems over two types of distributed sensing data, i.e., horizontally partitioned data and vertically partitioned data. The subproblems are solved using individual private data, and thus mHealth users can keep their private data locally and only upload (encrypted) intermediate results to the mHealth server for model training. Experimental results based on real datasets show that our scheme is highly efficient and scalable to a large number of mHealth users.

Index Terms—Private data analytics, mobile health, predictive model training, logistic regression.

1 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 biomedical sensors. The biomedical sensors can be manufactured to be light, durable, and comfortable at low cost and can sense a large variety of biomedical signals or physical activities, such as electrocardiogram, glucose concentration, breathing rate, pulse rate, blood pressure, peripheral oxygen saturation, and body motion [1], [2]. An example of such biomedical sensors is the "biostamp" designed by a company called MC10, which is quarter-size, waterproof, and breathable, and costs just tens of cents under batch production [3]. The sensed data can be transmitted to a remote mHealth server, which conducts analysis on the biomedical data and returns timely advices to the sensed subject. Health monitoring through biomedical sensors enables timely intervention and better management of individual health status, thus significantly improving healthcare quality.

Biomedical sensing data collected in health monitoring have attracted much research interest. First, the subjects of biomedical sensing include both patients and healthy people. The data of healthy people are not available in traditional healthcare because medical data are only collected when patients visit clinics. However, biomedical data from healthy people can be used as positive samples for training predictive models and will add important insights of disease prevention and prediction. Second, since biomedical sensors can monitor the human body day and night over a long time span, the data collected by biomedical sensors have much larger volume than traditional medical data. Data collected at this scale enable fine-grained diagnosis and treatment such as personalized medicine, and may largely improve healthcare quality and efficiency [4]. Due to the huge potential of biomedical sensing data in healthcare, researchers from the Institute of System Biology have initiated a project called 100K Wellness Project, which aims to intensely monitor $100,000$ healthy individuals and observe their physiology for 25 years [5]. It is envisioned that analysis on large-scale biomedical sensing data will reveal the earliest harbingers of killer diseases such as cancer and heart disease.

In this paper, we focus on logistic regression, a classic machine learning technique which is appropriate for predicting dichotomous outcomes and thus widely used for making decisions in medical diagnosis and prognosis [6]. For example, logistic regression can be used for calculating the probability that a patient will suffer cardiovascular disease [7], diabetes [8], and postpartum depression [9]; and it is also used for predicting the mortality probability in blunt trauma [10] and after a heart surgery [11]. Due to the diversity of human physiology, classifiers trained on individual datasets may not be robust over a wide range of input data. The availability of large-scale biomedical sensing data paves the way to collaborative learning [12], which overcomes the limitation by utilizing multiple user datasets with enough diversity. In collaborative learning, multiple individuals confide their data to a centralized party (e.g., a cloud server or a research institution, hereafter called mHealth server) as training samples [13]–[15]. The centralized party then constructs mathematical models based on the data. For mHealth applications, the collaborative learning may engage patients with the same disease, patients under similar treatment, or patients carrying certain genetic patterns. For ease of presentation, we use the term "Patient" to represent the subject of sensing, including both healthy persons and patients. Note that the

This work was partially supported by the U.S. National Science Foundation under grants CNS-1423165 and CNS-1409797. A preliminary version of this work has been published in IEEE GLOBECOM'15.

- *Y. Gong and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA. E-mail: {ymgong@, fang@ece.}ufl.edu.*
- *Y. Guo is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA. E-mail: richard.guo@okstate.edu.*

first character of the term is capitalized to remind readers of its broad meaning.

Although collaborative learning based on biomedical sensing data can be effective in predictive model training, it also raises serious privacy concerns. Medical data have always been private in nature. However, privacy issues in mHealth is especially prominent in multiple aspects. First, the mHealth server collects a wide range of health information including both physiological and physical activity data. While physiological data reflect health status of Patients and are private in nature, physical activity data may reveal sensitive information about lifestyles and activities of Patients. Second, mHealth devices usually collect user data continuously over a long time period, and thus the sensing data contain more private information than medical data collected in traditional clinic visits. Third, mHealth applications can be run by a wide range of parties. Thus the data may not only be learned by healthcare providers, but also insurance companies, diet advisers, athletic coaches or home-care providers. In such a setting, Patients may not trust the mHealth server with their private data. Hence, to incentivize mHealth users to contribute their data for model construction, we should guarantee their data privacy.

In this paper, we develop a privacy-preserving collaborative learning scheme that utilizes continuous sensing data from multiple Patients towards training logistic regression models in mHealth. The scenario we consider here is that the training samples are private while the resulting models are publicly available. We innovatively combine a distributed algorithm with a modified version of homomorphic encryption and give a scalable and practical solution for private model training in mHealth without an active third party. Unlike previous approaches, we leverage the intrinsic structure of the logistic regression model and decompose the collaborative learning problem into multiple subproblems that can be locally solved. An aggregate classifier is then computed by averaging locally trained parameters. The local training and averaging steps are repeated multiple rounds until the aggregate classifier converges. Specifically, we consider two different cases of distributed sensing data:

- *Horizontally partitioned data*: All Patients have a database of sensing data that are sensed by the same set of sensors. A typical setting is data collected through mobile health monitoring programs such as fitness tracking applications, where Patients' activities and sleep patterns are collected through a certain type of wearable devices.
- *Vertically partitioned data*: Each Patient only owns a few sensors and has a database sensed with partial sensors. With collaborative learning, we try to exploit these partial sensing data to find a common health pattern among the users. A typical setting is for the analysis of group therapy, where each Patient in the group senses her own data during every group meeting.

Our scheme is highly efficient and incurs low computational and communication 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 2. Then we develop distributed algorithms to decompose the logistic regression model in Section 3. We further present a secure summation protocol that enhances the privacy of the distributed algorithms in Section 4. Section 5 demonstrates experimental results and the performance analysis. We summarize related work in Section 6. Finally, Section 7 concludes the work.

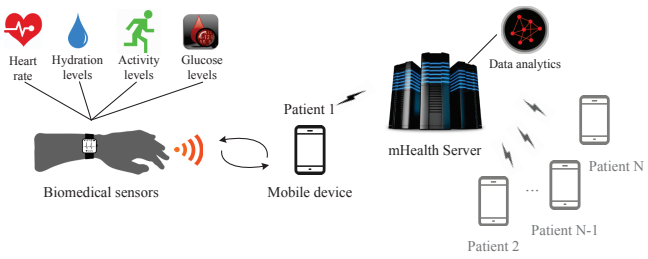


Fig. 1. Architecture of Model Training based on Biomedical Sensing Data.

2 System Model

In this section, we first outline the system architecture for private predictive modeling and describe the threat model and design goals. We then cover background on logistic regression and present the motivating scenarios in which private computation on Patient data is desirable.

2.1 System Architecture

We focus on patient-centered mHealth systems where Patients share their private sensing data with an mHealth server for model training. The system architecture 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. These 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. The task for the mHealth server is to construct a logistic regression model which enables computation of the outcome's probability given a new feature vector. The model needs to be collaboratively learned using datasets from multiple Patients.

In order to tune the logistic regression model based on the distributed datasets, we design an iterative algorithm which decomposes the original logistic regression model training problem into small subproblems. In each iteration, Patients use their own private data to construct intermediate local classifiers, which are aggregated later at the mHealth server. Since local classifiers are trained based on the data of each individual Patient, they may contain sensitive information about Patients. To prevent privacy leakage in local classifiers, we decompose the logistic regression problem in a way such that in each iteration, the mHealth server only needs to perform a simple average operation over local classifiers. We further protect the information of local classifiers by layering an efficient secure summation protocol onto the distributed algorithm so that the mHealth server learns nothing other than the aggregated result in each iteration. Individual Patient data are thus masked out in the aggregates which are safe to release.

2.2 Threat Model

We have the following security assumptions. The mHealth server is assumed to be Honest-but-Curious (HbC). On one hand, the mHealth server will honestly follow the protocol and is trustworthy to correctly compute predictive models. This is reasonable since it will be in the best interest of the mHealth server

to obtain an accurate and unbiased model. The mHealth server has no incentive to tamper intermediate aggregates or prevent/delay the convergence of the algorithms. On the other hand, the mHealth server is curious about the private feature vectors of Patients which they do not want to share. To this end, the mHealth server may (passively) attempt to infer private inputs of Patients or collude with some of the Patients to infer private information about other honest Patients.

We also assume that Patients are HbC. This means that Patients will faithfully follow the protocols, but they may collude with the mHealth server or some Patients to infer the inputs of other Patients. Nevertheless, we assume that only a small fraction of Patients will collude with the mHealth server. Note that it is possible that some Patients are not HbC and are incentivized to bias the computation results. However, when these Patients send largely biased data for model training, the uncommon data may be detected through signal processing techniques [16]. Thus, to avoid detection, they are assumed to only send slightly biased data, which are masked out in the aggregates of data from a large population of Patients and cannot have much influence on the accuracy of the computation results. Hence, we argue that in our setting HbC security is sufficient.

We do not consider outsider attacks because such attacks can be mitigated with system level protection and standard network security techniques. Specifically, we assume that the mobile device at each Patient is secure (i.e., the data stored at the mobile devices are protected from intrusion), and all the communication channels are reliable, encrypted and authenticated. Due to limited resources of biomedical sensors, lightweight cryptography schemes should be employed for data transmission from sensors to mobile devices, such as the one introduced in [17].

2.3 Design Goals

Our goal is to provide a practical privacy-preserving solution for real world model training under aforementioned security assumptions. To this end, we identify three key properties for a practical privacy-preserving solution.

- **Privacy**. Since the mHealth server is not trusted to learn the training samples, they should be kept locally at the Patient's side. *How can we design a collaborative learning scheme based on distributed data?* We will answer this question with distributed approaches which iteratively train and aggregate 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 could reveal sensitive information about Patients [18]. *How can we ensure no private information is leaked during the aggregation process?*
- **Scalability**. Our scheme should be scalable to a large number of mHealth users so that the training samples provide sufficient diversity for training robust classifiers. The main factor that influences scalability is the computational complexity. Considering the limited computational resources and battery lifetime of mobile devices, we need to keep the computational and communication overhead at the Patient side low even with a large number of participating Patients.
- **Efficiency**. Our training scheme must be efficient for time-series sensing data. In mHealth monitoring, samples are continuously generated over multiple time periods. The predictive model may be periodically 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).

2.4 Logistic Regression

Logistic regression is a classic machine learning technique that is commonly used in predicting dichotomous outcomes in medical diagnosis and prognosis [6]. Here we briefly discuss the basics of logistic regression. Without loss of generality, we focus on binary class logistic regression, but our solution is directly applicable to the case of k-class logistic regression.

Consider a supervised learning task with a set of labeled training samples $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ denotes a feature vector and $y_i \in \{-1, +1\}$ denotes the corresponding binary class label. The ℓ_1 regularized logistic regression problem [19] is defined as

$$\min \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i\left(\mathbf{w}^T\mathbf{x}_i + v\right)\right)\right) + \lambda \left\|\mathbf{w}\right\|_1, \quad (1)$$

where the weight vector $\mathbf{w} \in \mathbb{R}^n$ and intercept $v \in \mathbb{R}$ are the parameters of the logistic regression model, and $\lambda > 0$ is the regularization parameter.

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

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

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

The resulting classifier can predict the class label of a new feature vector, and thus is particularly suitable for disease state prediction (healthy or unhealthy) and decision making (yes or no) in medical diagnosis and prognosis. For instance, in [8], Tabaei and Herman conduct 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), and random capillary plasma glucose level (RPG). Each feature vector is associated with a label y_i, 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, when FPG $\geq$ 140 mg/dl or 2-h PG $\geq$ 200, $y_i = 1$; otherwise, $y_i = -1$. The mHealth server collects data from $1,032$ Patients and uses 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 $\mathbf{x}$, the classifier can predict the probability that FPG $\geq$ 140mg/dl or 2-h PG $\geq$ 200 according to (2) and (3).

Although having many benefits in medical field, logistic regression poses significant threats to user privacy since it involves the usage of private sensing data such as blood level, activity, and age. Therefore, a scheme to preserve user privacy while not sacrificing utility of the medical sensing data is needed.

3 Private Predictive Model Training Via Distributed Computation

In this section, we describe a practical privacy-preserving scheme that enables collaborative model learning over distributed data. During collaborative model training, each Patient contributes a set of training samples. Each training sample is associated with a label ("+1" or "-1" for binary class). The training samples are considered private as they may reveal sensitive information such as health status and unusual activities of individuals. Our scheme is based on an algorithm called alternating direction method of multipliers (ADMM). The algorithm provides a possible way to decompose the logistic regression model into smaller subproblems that can be locally computed. In this section, we first give some background on ADMM. Then we use ADMM to design privacy-aware distributed algorithms that solve logistic regression under two cases of Patient data: horizontally partitioned data and vertically partitioned data, which correspond to different application scenarios.

3.1 Basics of ADMM

In the following, we describe the basics of ADMM. ADMM is a distributed algorithm that solves a large-scale optimization problem by decomposing it into smaller subproblems that are easier to solve. ADMM is first introduced by Glowinski, Marroco, Gabay, and Mercier [20], [21] in 1976 and has found applications in many areas since then [22]. The algorithm solves problems in the following form:

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

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

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

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

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

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:

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

2) z-minimization with x and y fixed:

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

3) Dual variable y update:

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

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 theorem, whose proof can be found in [23].

Theorem 1 ([23]). *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.

3.2 Horizontally Partitioned Data

The term "horizontally partitioned data" is initially used in databases where the data are partitioned based on rows. In our mHealth scenario, Patients have the same set of biomedical sensors and each Patient generates sensing data with the same number of features, as shown in Fig. 2a. Each Patient stores several rows of sensing data with each row containing the sensing results collected in a single sampling period. A motivating scenario that generates such data is mobile health monitoring for diabetes management. Consider that a research institute wants to study the risk factors that influence the glucose level and construct a predictive model that predicts whether the glucose level will be normal or not given these risk factors. The institute recruits a group of Patients for its study. As part of the study, a Patient wears a mobile device provided by the institute that continuously monitors factors including medication (e.g., insulin intake), physical activity (e.g., light exercise), food intake (e.g., carbohydrates), and other biological (e.g., dawn phenomenon) and environmental (e.g., altitude) factors. The Patient also records his/her 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 with a safety threshold. Both feature vectors and labels are sent to the institute, who then trains a model that predicts whether the blood glucose level is above or below the safety threshold. Such a model will help diabetics better monitor their blood glucose levels and reduce the frequencies of unpleasant blood tests.

In the case of horizontally partitioned data, Patients are equipped with the same set of sensors and each Patient obtains a set of feature vectors. Specifically, each Patient i has a local set of training samples $\mathcal{D}_i:=\{(\mathbf{a}_{ij},b_{ij}),j=1,\ldots,m_i\}$, where $\mathbf{a}_{ij}\in\mathbb{R}^n$ is a feature vector, $b_{ij}\in\{-1,+1\}$ is the corresponding label of the outcome variable, and m_i is the number of training samples owned by Patient i. The label is owned by the Patient and assumed to be private. The ℓ_1 regularized logistic regression problem becomes the following: Given a set of labeled training samples $\cup_{i=1}^N\mathcal{D}_i$ from N Patients, solve

$$\min\sum_{i=1}^{N}\sum_{j=1}^{m_i}\log\left(1+\exp\left(-b_{ij}\left(\mathbf{w}^T\mathbf{a}_{ij}+v\right)\right)\right)+\lambda\left\|\mathbf{w}\right\|_1, \tag{10}$$

where $\mathbf{w}\in\mathbb{R}^n, i=1,...,N$ and $v\in\mathbb{R}$.

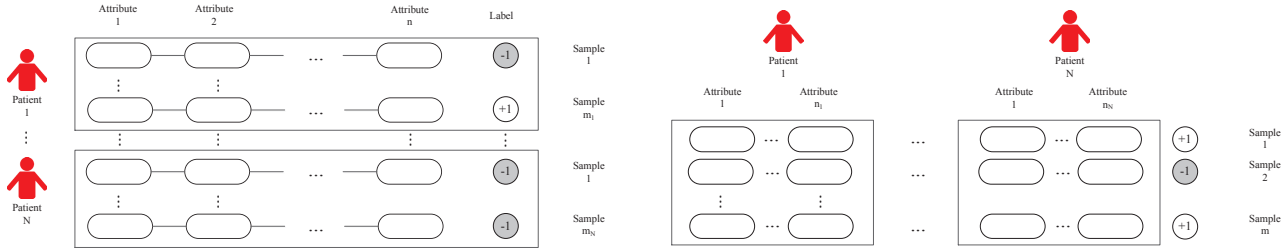


Fig. 2. Illustration of (a) Horizontally and (b) Vertically Partitioned Data.

The above problem cannot be directly solved 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 $\{(\mathbf{w}_i, v_i)\}, i = 1, \ldots, N$ and reformulate the optimization problem as

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(\mathbf{w}_i^T\mathbf{a}_{ij}+v_i\right)\right)\right) + \lambda\left\|\mathbf{w}\right\|_1 \\ \text{s.t.} \quad & \mathbf{w}_i = \mathbf{w}, v_i = v, i = 1, \ldots, N. \end{aligned} \tag{11}$$

It is obvious that the new problem (11) is equivalent to the original problem (10). Note that the objective function in the problem (11) is now separable over two sets of variables $\{(\mathbf{w}_i, v_i), i = 1, \ldots, N\}$ and $(\mathbf{w}, v)$. We can view $(\mathbf{w}_i, v_i)$ as the copy of regression parameters at Patient i 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 into several subproblems. For simplicity of notation, we define $\alpha := \{(\mathbf{w}, v)\}$ and $\beta := \{(\mathbf{w}_i, v_i), i = 1, \ldots, N\}$. Following the framework of ADMM, we formulate the augmented Lagrangian of (11) as

$$\begin{aligned} L_\rho(\alpha, \beta, \gamma) = & \sum_{i=1}^{N}\sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(\mathbf{w}_i^T\mathbf{a}_{ij}+v_i\right)\right)\right) \\ & + \lambda\left\|\mathbf{w}\right\|_1 + \sum_{i=1}^{N}\left((\mathbf{w}_i-\mathbf{w})^T\boldsymbol{\gamma}_{i,w} + \gamma_{i,v}(v_i - v)\right) \\ & + \sum_{i=1}^{N}(\rho/2)\left((\mathbf{w}_i-\mathbf{w})^T(\mathbf{w}_i-\mathbf{w}) + (v_i-v)^2\right), \end{aligned} \tag{12}$$

where $\gamma := \{(\boldsymbol{\gamma}_{i,w}, \gamma_{i,v}), i = 1, \ldots, N\}$ are the dual variables corresponding to the constraints in (11).

We 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:

$$\begin{aligned} \min_{\alpha} \quad & \lambda\left\|\mathbf{w}\right\|_1 + (\rho N/2)\mathbf{w}^T\left(\mathbf{w} - 2\overline{\mathbf{w}}^k - 2\overline{\boldsymbol{\gamma}}_w^k/\rho\right) \\ & + (\rho N/2)v\left(v - 2\overline{v}^k - 2\overline{\gamma}_v^k/\rho\right), \end{aligned} \tag{13}$$

where the overline notation $\overline{(\cdot)}$ 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 [24]. Specifically, the optimal solution is given by

$$\begin{aligned} \mathbf{w}^{k+1} := & \left[\overline{\mathbf{w}}^k + \overline{\boldsymbol{\gamma}}_w^k/\rho - (\lambda/\rho N)\right]_+ \\ & - \left[-\overline{\mathbf{w}}^k - \overline{\boldsymbol{\gamma}}_w^k/\rho - (\lambda/\rho N)\right]_+ \end{aligned} \tag{14a}$$

$$v^{k+1} := \overline{v}^k + \overline{\gamma}_v^k/\rho, \tag{14b}$$

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:

$$\begin{aligned} \min_{\beta} \quad & \sum_{i=1}^{N}\sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(\mathbf{w}_i^T\mathbf{a}_{ij}+v_i\right)\right)\right) \\ & + \sum_{i=1}^{N}(\rho/2)\mathbf{w}_i^T(\mathbf{w}_i - 2\mathbf{w}^{k+1} + 2\boldsymbol{\gamma}_{i,w}^k/\rho) \\ & + \sum_{i=1}^{N}(\rho/2)v_i(v_i - 2v^{k+1} + 2\gamma_{i,v}^k/\rho), \end{aligned} \tag{15}$$

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

$$\begin{aligned} \min_{\beta_i} \quad & \sum_{j=1}^{m_i} \log\left(1+\exp\left(-b_{ij}\left(\mathbf{w}_i^T\mathbf{a}_{ij}+v_i\right)\right)\right) \\ & + (\rho/2)\mathbf{w}_i^T(\mathbf{w}_i - 2\mathbf{w}^{k+1} + 2\boldsymbol{\gamma}_{i,w}^k/\rho) \\ & + (\rho/2)v_i(v_i - 2v^{k+1} + 2\gamma_{i,v}^k/\rho). \end{aligned} \tag{16}$$

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.

After we obtain α^{k+1} and β^{k+1}, the dual update is as follows:

$$\boldsymbol{\gamma}_{i,w}^{k+1} := \boldsymbol{\gamma}_{i,w}^k + \rho\left(\mathbf{w}_i^{k+1} - \mathbf{w}^{k+1}\right), \tag{17a}$$

$$\gamma_{i,v}^{k+1} := \gamma_{i,v}^k + \rho\left(v_i^{k+1} - v^{k+1}\right). \tag{17b}$$

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 Patient i will learn the global optimal classifiers $\mathbf{w}$ and v without sending his/her local training set to others. Therefore, this system can preserve user privacy without sacrificing the utility of the learning function.

Algorithm 1 Distributed Algorithm for Horizontally Partitioned Data

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 $\mathbf{v}^{k+1}$ according to (14) 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 (16) independently using his/her own training set and then updates independently the dual variables according to (17).
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

3.3 Vertically Partitioned Data

The term "vertically partitioned data" refers to partitioning data based on columns in databases. In vertically partitioned datasets, each row represents data sampled at the same time slot or under the same context, and the data in each row are collaboratively sensed by all Patients as shown in Fig. 2b. Instead of exploiting the diversity of individuals for robustness as in the horizontally partitioned case, we try to involve more features of sensing data. We assume that Patients own disjoint subsets of sensors, and we aim to fit models with sensing data from the union of all these sensors. A motivating scenario is the analysis of group therapy. Consider a therapy group where a therapist treats a group of Patients. After each group meeting, the therapist will evaluate the group meeting as effective ("+1") or ineffective ("-1"), which is the label of this meeting. Each participant of the group is equipped with sensors to sense his/her own biological and emotional status during the group meeting. The feature vector of a group meeting is the sensing data from all Patients. A series of group meeting will be held during some period, resulting in a vertically partitioned distributed dataset.

The vertically partitioned case is particularly useful when we need to monitor the data of all members in a group for group effect evaluation, as in the group therapy case. It is also very useful in a high dimensional data setting where the number of features (i.e., sensed metrics or risk factors of a disease) is very large. In this case, it would be a great commitment for an individual to use all sensors on his/her body, especially under a continuous sensing environment, because it is cumbersome and inconvenient. Thus an individual may hesitate to participate in such projects. Moreover, in many cases the training data are by-products of other health monitoring programs where Patients only use a small set of sensors that are closely related to their own health statuses.

In the case of vertically partitioned data, each Patient i is equipped with a subset of sensors and obtains partial feature vectors over a specified time interval. Specifically, each Patient i poses a set of training samples $\hat{\mathcal{D}}_i := \{(\hat{\mathbf{a}}_{ij}, \hat{b}_j), j = 1, \ldots, m\}$, where $\hat{\mathbf{a}}_{ij} \in \mathbb{R}^{n_i}$ is a partial feature vector, $\hat{b}_j \in \{-1, 1\}$ is the corresponding label of the outcome variable, m is the number of training samples owned by each user, and $\sum_{i=1}^N n_i = n$. As with other papers [25] in literature, the labels $\{\hat{b}_j, j = 1, \ldots, m\}$ are assumed to be known by all Patients and not private. Then, the ℓ_1 regularized logistic regression problem becomes the following: Given a set of labeled training samples $\cup_{i=1}^N \hat{\mathcal{D}}_i$, solve

$$\min \sum_{j=1}^m \log\left(1 + \exp\left(-\hat{b}_j\left(\sum_{i=1}^N \mathbf{w}_i^T \hat{\mathbf{a}}_{ij} + v\right)\right)\right) + \lambda \sum_{i=1}^N \|\mathbf{w}_i\|_1, \quad (18)$$

where $\mathbf{w}_i \in \mathbb{R}^{n_i}, i = 1, \ldots, N$ and $v \in \mathbb{R}$.

To solve the above problem with ADMM, we first introduce a set of auxiliary variables $\{z_{ij}, i = 1, \ldots, N, j = 1, \ldots, m\}$ and reformulate the optimization problem as

$$\min \sum_{j=1}^m \log\left(1 + \exp\left(-\hat{b}_j\left(\sum_{i=1}^N z_{ij} + v\right)\right)\right) + \lambda \sum_{i=1}^N \|\mathbf{w}_i\|_1,$$
$$\text{s.t. } \mathbf{w}_i^T \hat{\mathbf{a}}_{ij} - z_{ij} = 0, i = 1, \ldots, N, j = 1, \ldots, m. \quad (19)$$

It is obvious that the new problem (19) is equivalent to the original problem (18). The objective function now is separable over two sets of variables $\{(v, z_{ij}), i = 1, \ldots, N, j = 1, \ldots, m\}$ and $\{\mathbf{w}_i, i = 1, \ldots, N\}$.

In the following, we demonstrate that through these auxiliary variables the problem can be decomposed. For simplicity of notation, we define $\hat{\alpha} := \{\mathbf{w}_i, i = 1, \ldots, N\}$ and $\hat{\beta} := \{(v, z_{ij}), i = 1, \ldots, N, j = 1, \ldots, m\}$. Following the framework of ADMM, we formulate the augmented Lagrangian as

$$\begin{aligned} L_\rho(\hat{\alpha}, \hat{\beta}, \hat{\gamma}) = &\sum_{j=1}^m \log\left(1 + \exp\left(-\hat{b}_j\left(\sum_{i=1}^N z_{ij} + v\right)\right)\right) \\ &+ \lambda \sum_{i=1}^N \|\mathbf{w}_i\|_1 + \sum_{j=1}^m \sum_{i=1}^N \gamma_{ij}\left(w_i^T \hat{\mathbf{a}}_{ij} - z_{ij}\right) \\ &+ \sum_{j=1}^m \sum_{i=1}^N (\rho/2)\left(\mathbf{w}_i^T \hat{\mathbf{a}}_{ij} - z_{ij}\right)^2, \end{aligned}$$

where $\hat{\gamma} := \{\hat{\gamma}_{ij}, i = 1, \ldots, N, j = 1, \ldots, m\}$ are the dual variables corresponding to constraints (19).

Our algorithm works as follows. At the $(k+1)$-th iteration, the $\hat{\alpha}$-minimization step involves solving the following problem for each Patient i in parallel:

$$\min \lambda \|\mathbf{w}_i\|_1 + (\rho/2) \sum_{j=1}^m \mathbf{w}_i^T \hat{\mathbf{a}}_{ij}\left(\mathbf{w}_i^T \hat{\mathbf{a}}_{ij} - 2z_{ij}^k + 2\hat{\gamma}_{ij}^k/\rho\right). \quad (20)$$

After obtaining $\hat{\alpha}^{k+1}$ from the $\hat{\alpha}$-minimization step, the $\hat{\beta}$-minimization step consists of solving the following:

$$\begin{aligned} &\min \sum_{j=1}^m \log\left(1 + \exp\left(-\hat{b}_j\left(\sum_{i=1}^N z_{ij} + v\right)\right)\right) \quad (21) \\ &+ (\rho/2) \sum_{j=1}^m \sum_{i=1}^N z_{ij}\left(z_{ij} - 2(\mathbf{w}_i^{k+1})^T \hat{\mathbf{a}}_{ij} - 2\hat{\gamma}_{ij}^k/\rho\right). \end{aligned}$$

The $\hat{\beta}$-minimization problem can be further simplified as follows. Let $\overline{z}_j$ denote the average of z_{ij} across all i. The $\hat{\beta}$-update problem can be rewritten as

$$\min \quad \sum_{j=1}^{m} \log\left(1+\exp\left(-\hat{b}_j\left(N\overline{z}_j+v\right)\right)\right) \tag{22a}$$

$$+(\rho/2)\sum_{j=1}^{m}\sum_{i=1}^{N} z_{ij}\left(z_{ij}-2(\mathbf{w}_i^{k+1})^T\hat{\mathbf{a}}_{ij}-2\hat{\gamma}_{ij}^k/\rho\right)$$

$$\text{s.t.} \quad \overline{z}_j=(1/N)\sum_{i=1}^{N} z_{ij} \tag{22b}$$

Note that in the above problem, minimizing over $z_{ij}, \forall i$ with $\overline{z}_j$ fixed has the solution

$$\begin{aligned} z_{ij} =&(\mathbf{w}_i^{k+1})^T\hat{\mathbf{a}}_{ij}+\hat{\gamma}_{ij}^k/\rho+\overline{z}_j \\ &-(1/N)\sum_{i=1}^{N}\left(\hat{\gamma_{ij}}^k/\rho+(\mathbf{w}_i^{k+1})^T\hat{\mathbf{a}}_{ij}\right). \end{aligned} \tag{23}$$

Therefore, Problem (22) can be computed by solving the following unconstrained optimization problem:

$$\begin{aligned} \min \sum_{j=1}^{m}\Bigg(&\log\left(1+\exp\left(-\hat{b}_j\left(N\overline{z}_j+v\right)\right)\right)+(\rho N/2)\overline{z}_j^2 \\ &-\rho\overline{z}_j\sum_{i=1}^{N}\left(\hat{\gamma}_{ij}^k/\rho+(\mathbf{w}_i^{k+1})^T\hat{\mathbf{a}}_{ij}\right)\Bigg) \end{aligned} \tag{24}$$

and then applying (23) to obtain z_{ij}.

By substituting (23) for z_{ij}^{k+1} in the dual update equation gives

$$\hat{\gamma}_{ij}^{k+1}:=\rho\left((1/N)\sum_{i=1}^{N}\left(\hat{\gamma}_{ij}^k/\rho+(\mathbf{w}_i^{k+1})^T\hat{\mathbf{a}}_{ij}\right)-\overline{z}_j^{k+1}\right), \tag{25}$$

which does not depend on i. Therefore, the dual variables $\hat{\gamma}_{ij}^{k+1}, i=1,\ldots,N$ are all equal and can be replaced by a single dual variable $\hat{\gamma}_j^{k+1}$.

In summary, by substituting $\hat{\gamma}_j$ and (23) into the $\hat{\alpha}$-minimization, $\hat{\beta}$-minimization, and dual variable update equation, our final algorithm consists of the following iterations:

$$\begin{aligned} \mathbf{w}_i^{k+1}:=&\operatorname*{argmin}_{\mathbf{w}_i}\lambda\left\|\mathbf{w}_i\right\|_1+(\rho/2)\sum_{j=1}^{m}\left(\mathbf{w}_i^T\hat{\mathbf{a}}_{ij}\right)^2 \\ &-\rho\sum_{j=1}^{m}\mathbf{w}_i^T\hat{\mathbf{a}}_{ij}^T\left((\mathbf{w}_i^T)^k\hat{\mathbf{a}}_{ij}+\overline{z}_j^k+\frac{\hat{\gamma}_j^k}{\rho}-\frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}_i^T)^k\hat{\mathbf{a}}_{ij}\right) \end{aligned} \tag{26}$$

$$\begin{aligned} \hat{\beta}^{k+1}:=&\operatorname*{argmin}_{\overline{z},v}\sum_{j=1}^{m}\Bigg(\log\left(1+\exp\left(-\hat{b}_j\left(N\overline{z}_j+v\right)\right)\right) \\ &-\hat{\gamma}_j^kN\overline{z}_j+\frac{\rho N}{2}\overline{z}_j^2-\rho\overline{z}_j\sum_{i=1}^{N}(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}\Bigg) \end{aligned} \tag{27}$$

$$\hat{\gamma}_j^{k+1}:=\hat{\gamma}_j^k+\rho\left(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}-\overline{z}_j^{k+1}\right). \tag{28}$$

The entire procedures of our algorithm are described in Algorithm 2. At the end of the algorithm, Patients will learn the global optimal regression parameters $\mathbf{w}$ and v without disclosing their local training set to others.

Algorithm 2 Distributed Algorithm for Vertically Partitioned Data

1: Initialization: $k\leftarrow 0$, $(\mathbf{w}_i^T)^0\leftarrow 0$, $\overline{z}_j^0\leftarrow 0$, and $\hat{\gamma}_j^0\leftarrow 0$.
2: **repeat**
3: Each Patient i solves the per-Patient subproblem (26) independently using his/her own training set to obtain the optimal solution $\mathbf{w}_i^{k+1}$ and then sends $\{(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}, j=1,\ldots,m\}$ to the mHealth server.
4: After gathering $\{(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}, j=1,\ldots,m\}$ from all Patients $i=1,\ldots,N$, the mHealth server averages them to obtain $(1/N)\sum_{i=1}^{N}(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}, j=1,\ldots,m$. Then it updates v^{k+1} and $\{\overline{z}_j^{k+1}, j=1,\ldots,m\}$ according to (27), and dual variables $\{\hat{\gamma}_j^{k+1}, j=1,\ldots,m\}$ according to (28).
5: The mHealth server broadcasts $\overline{z}_j^{k+1}$, $\hat{\gamma}_j^{k+1}$, and $(1/N)\sum_{i=1}^{N}(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}$ to all Patients.
6: $k\leftarrow k+1$
7: **until** Convergence criteria is met

4 Private Aggregation of Local Regression Parameters

In this section, we describe a secure summation protocol that enhances the privacy of the distributed algorithms. The protocol computes the sum over encrypted values such that only the sum is learned by the mHealth server.

4.1 Private Computation at the mHealth Server

In the distributed algorithm for horizontally partitioned data, each Patient i sends his/her local 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 (Line 4, Algorithm 1). However, these local regression parameters are trained on individual private data and may leak sensitive information about Patients [18]. We observe that in each iteration, the mHealth server only needs to know the average of these local optimal solutions, i.e., $\overline{\mathbf{w}}^k$, $\overline{v}^k$, $\overline{\boldsymbol{\gamma}}_w^k$, and $\overline{\gamma}_v^k$. Similar observation can be made for the distributed algorithm for vertically partitioned data, where the mHealth server gathers $\{(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}, j=1,\ldots,m\}$ from all Patients $i=1,\ldots,N$ but only needs to know the averages $(1/N)\sum_{i=1}^{N}(\mathbf{w}_i^T)^{k+1}\hat{\mathbf{a}}_{ij}, j=1,\ldots,m$ (Algorithm 2, Line 4). Hence, the privacy issues in both algorithms can be mitigated if the mHealth server can calculate the average (or sum) without knowing the individual values.

We will first describe a naive approach that enables secure summation over distributed private values but leaks private values under collusion attacks. Then we will present a solution that mitigates such attacks with low computational and communication overhead. For simplicity we only describe how the mHealth server can obtain $\overline{v}^k$ from distributed values $v_i^{k+1}, i=1,\ldots,N$ without learning them. Without loss of generality, we assume that v_i^{k+1} is an integer. Averaging over other types of distributed values (i.e., real numbers or vectors) can be calculated in a similar way: (i) When v_i^k 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 [26]; (ii) Averaging vectors can be treated as aggregating scalars at each component of the vectors. When the context is clear, we omit the superscript k and $k+1$.

Naive approach. A naive solution to averaging distributed private values is the secure summation protocol proposed by

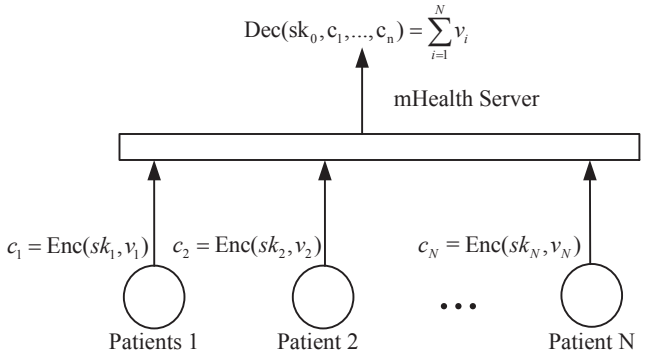


Fig. 3. Aggregation of Private User Data.

Clifton et. al. [27]. Using their protocol, patients are arranged in a unidirectional ring with one patient acting as the protocol initiator. The protocol initiator selects a random number and adds the number to his/her own data, then the sum is passed along the ring, with Patients along the ring adding their own data to the sum. When the protocol initiator receives the sum again, he/she subtracts the random number from the sum and obtain the accurate sum of all Patient's data. The average can be directly computed by dividing the sum by N. Since the values passed between Patients are masked by a random value, which is only known by the protocol initiator, these values are kept private. However, this approach is not secure against collusion: If the two neighbors of a Patient collude, they can infer the private value of the Patient. Moreover, this protocol requires Patients to interact with each other whenever a sum needs to be calculated, which is undesirable for computation over multiple iterations as required in our algorithms.

Modified approach. To overcome the two aforementioned shortcomings, we use a homomorphic approach that is robust against collusion attacks and highly efficient for computation over multiple iterations [28]. With this approach, secure summation can be achieved without any active trusted third-parties. Moreover, this approach has low amortized computational overhead and is thus efficient for our iterative algorithms. The overview of this approach is shown in Fig. 3. At the beginning of the aggregation process, each Patient i has a secret key sk_i and the mHealth server has a secret key sk_0, where $\sum_{i=0}^{N} sk_i = 0$. The Patient encrypts his/her private data v_i as $v_i + sk_i$ and sends the ciphertext to the mHealth server. The mHealth server sums all the ciphertext and decrypts the sum as $\sum_{i=1}^{N} v_i = \sum_{i=1}^{N}(v_i + sk_i) - sk_0$. Since the mHealth server does not know the secret values of sk_i, the individual ciphertexts are meaningless random numbers from the view of the mHealth server. This scheme prevents collusion attack because each private value is randomized by a separate secret key and will only be revealed if the mHealth server colludes with all other Patients, which is unlikely.

In order to guarantee that $\sum_{i=0}^{N} sk_i = 0$, secret keys should be collaboratively generated in every iteration. This process requires an active trusted third party and is not practical for our iterative algorithms. To overcome this challenge, we rely on a hash function H that maps an integer to an appropriate mathematical group. In the kth iteration of our algorithms, each Patient i computes $H(k)^{sk_i}$ and the mHealth server computes $H(k)^{sk_0}$. From $\sum_{i=0}^{N} sk_i = 0$, we have $\prod_{i=0}^{N} H(k)^{sk_i} = 1$, which can be leveraged to generate secret keys without interactive communication in each iteration. We summarize the protocol below.

Let $\mathbb{G}$ denote a cyclic group of prime order p for which Decisional Diffie-Hellman problem is hard. Let $H : \mathbb{Z} \to \mathbb{G}$ 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, Patient i encrypts his/her 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 $P = H(k)^{sk_0} \prod_{i=1}^{n} c_i = H(k)^{\sum_{i=0}^{n} sk_i} \cdot g^{\sum_{i=1}^{n} v_i} = g^{\sum_{i=1}^{n} v_i}$. The sum of v_i can then be calculated by computing the discrete log of P base g.

The scheme allows untrusted mHealth server to periodically estimate the sum $\sum_{i=1}^{N} v_i$ without knowing each individual value of $v_i, i = 1, 2, \ldots, N$. The average $\overline{v}$ can then be readily calculated by dividing the sum by the total number of Patients N. Since Patients do not need to communicate with each other for sharing secrets after the initial key generation process, we only require a passive trusted authority during initialization. Hence, the computational overhead of secret keys does not increase with the iteration process, achieving low amortized computational overhead.

The computational overhead for the aggregation process in each iteration comes from two parts: encryption and secret key generation. Encryption operation in the construction includes one hash, one multiplication in a Diffie-Hellman group, and two modular exponentiations. The two modular exponentiations takes much longer than other operations and thus dominate the running time. According to the benchmarking report of eBACs project [29], 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 computational overhead for the Patients in mHealth applications.

4.2 Provable Privacy

We prove the privacy of our approach from the following two aspects:

First, we show that our computation protocol leaks no information beyond the intermediate and final aggregated regression parameters. We note that the homomorphic approach we use in our scheme is "aggregator oblivious" in the sense that the aggregator (i.e., mHealth server) learns only the sum for each time period and nothing more. Detailed proof of this property can be found in [28]. Basically, their proof is based on the assumption that the Decisional Diffie-Hellman problem is computationally infeasible for probabilistic polynomial-time adversaries. The "aggregator oblivious" property of the homomorphic approach guarantees that the mHealth server cannot learn any unintended information other than what can be deduced from its auxiliary knowledge and the revealed computation results. From the mHealth server's view, the input data of the aggregation protocol, i.e., Patients' intermediate

regression parameters, are indistinguishable from data uniformly chosen at random from the plaintext space.

Second, We show the information leakage during the iterations of our algorithms is bounded. Intuitively, we can view the sum of local parameters revealed in each iteration as "global" information and thus privacy-preserving in common practice. We can provide a strong privacy guarantee, (ϵ, δ)-differential privacy (DP) [30], [31], which ensures that the privacy risk of a user does not substantially increase if the user participates in collaborative learning despite of the auxiliary knowledge of adversaries. Formally speaking,

Definition 1. $\forall \epsilon, \delta \geq 0$, a randomized algorithm $\mathcal{F}$ gives (ϵ, δ)-DP if for any two datasets D_1 and D_2 which differ in only one element, and $\forall O \subseteq \text{range}(\mathcal{F})$, the following inequality holds:

$$\ln \frac{\Pr[\mathcal{F}(D_1) \in O] - \delta}{\Pr[\mathcal{F}(D_2) \in O]} \leq \epsilon. \tag{29}$$

Here the parameter ϵ bounds the ratio of probability distributions of two datasets differing on at most one element, and δ relaxes the strict relative shift at events that are not especially likely.

Most solutions that achieve differential privacy are based on perturbing the response with additive noise [30], [32] or perturbing the computation with external randomness [18], [33]. This can also be achieved in our scheme. Note that we want to add the proper amount of noise to the aggregated result rather than each individual local parameter, because the noise for the the latter case would be larger under the same privacy requirement. Meanwhile, we need to ensure that the aggregated results are perturbed before they are decrypted by the mHealth server. A feasible way to achieve this is to introduce a proxy which can be another server at the mHealth service provider, and let the server and the proxy collaboratively add noise before decrypting the aggregated results, as proposed in [28]. Since the aggregation process in our algorithms is repeated iteratively, the noise added in each iteration would accumulate. However, due to the good convergence properties of our algorithms (usually converge within a few tens of iterations), the total added noise can be controlled at a low level, ensuring the accuracy of the final results.

Furthermore, we note that the process of aggregation have already incorporated randomness, thus providing certain privacy protection itself. In fact, Duan [34] has provided a rigorous proof that differential privacy can be achieved by aggregating vectors from a large number of entities under certain constraints, as summarized in the following theorem:

Theorem 2 ([34]). *Let f be the sum of N n-vectors $w_i, i = 1, \ldots, N$, $w_i \in [0,1]^n$. Assuming $w_1, w_2, \ldots, w_N$ are i.i.d. with $E[w_i] = \tau, E[w_i w_i^T] - \tau\tau^T = V < \infty$, the summation is (ϵ, δ)-DP if N is sufficiently large and*

$$\lambda_{min}(V) > \frac{2n^2 \log(2n/\delta)}{(N-1)\epsilon^2}, \tag{30}$$

where $\lambda_{min}(V)$ is the smallest eigenvalue of matrix V.

This theorem provides a theoretic basis for achieving differential privacy in the aggregation process. Even if for a given set of (ϵ, δ), the constraint (30) is not satisfied, i.e., the aggregation process can not provide enough privacy protection, we may still achieve (ϵ, δ)-DP with the perturbation approach, and the amount of noise required in the perturbation approach may be further reduced with Theorem 2.

5 Performance Evaluation

In this section, we evaluate the performance of our approach based on real-world datasets. All the simulations are conducted in MATLAB using a notebook with 1.6GHz CPU and 4G memory.

To provide benchmarks for the performance of our distributed approach, we compare it with the following two baselines:

- *Centralized approach*: In this approach, the mHealth server has access to all Patient data and solves the logistic regression problem centrally. Although this algorithm can obtain the optimal performance, it violates Patient privacy and thus is privacy-oblivious.
- *Local approach*: In this approach, each Patient trains the logistic regression model solely based on his/her own local data. Since the performance of the model highly depends on the size of training set, the local approach has lower performance than the centralized approach. In other words, the local approach protects Patient privacy at the cost of utility or accuracy.

5.1 Results on Activity Recognition Task (Horizontally Partitioned Dataset)

We first demonstrate the performance of our distributed approach for horizontally partitioned data (Algorithm 1). We test our approach on the dataset for the physiological data modeling contest at the International Conference on Machine Learning in 2004 [35]. The dataset was collected from users using BodyMedia wearable body monitors. We use a subset of the dataset to classify two states of activities, which includes 4413 positive samples (context 1) and 98172 negative samples (context 2). Each sample contains 9 dimensional physiological data and 3 characteristics (denoted as "char" 1, "char" 2, and sex) of the users. Thus, we construct a 102585×12 matrix from the monitoring data. The label for each row is the context of the user when the sample is collected: When the user is under context 1, we label it as 1; otherwise, we label it as -1. We aim to train a logistic regression model that predicts the label based on a new sample. In each experimental trial, we randomly select 14000 training samples (4000 positive samples, 10000 negative samples) and 1413 testing samples (413 positive samples, 1000 negative samples). The test error rates of algorithms are averaged results of 10 experimental trials.

We implement our distributed algorithm and observe good convergence properties for different numbers of Patients N. Since the convergence properties for different N are similar, we only demonstrate the convergence results for $N = 1000$. The convergence property of our distributed algorithm is depicted in Fig. 4, which shows the change of the logistic regression parameters w.r.t. the iteration number k. The x-axis of the plot represents the number of iterations k, and the y-axis of the plot represents the norm of the distance between the global optimal regression parameters and the intermediate regression parameters in each iteration. We can see from the figure that the logistic regression parameters obtained by our approach converges fast (around 40 iterations) to the optimal ones. To demonstrate the accuracy loss of the distributed approach w.r.t. the centralized approach, we show in Fig. 5 the change of the objective function value w.r.t. iteration number k. The solid line indicates the objective value obtained by our approach, and the dashed line denotes the global optimal objective value obtained by the centralized approach. As shown in the figure, the objective value of our approach decreases very fast

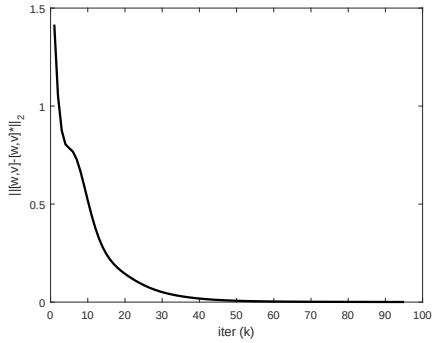


Fig. 4. Convergence of the logistic regression parameters obtained from our distributed approach on the horizontally partitioned dataset.

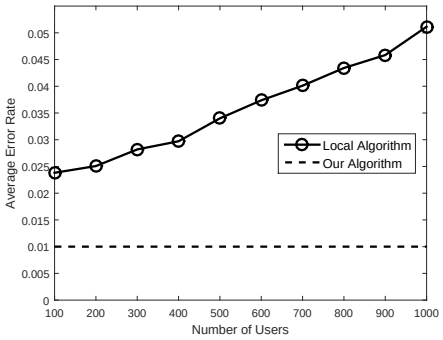


Fig. 6. Testing errors for our distributed approach and the local approach on the horizontally partitioned dataset.

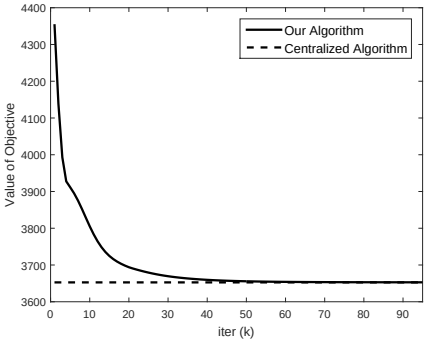


Fig. 5. Comparison of the objective of our approach (solid line) and the optimal objective (dashed line) on the horizontally partitioned dataset.

in the first few iterations and finally approaches the optimal value after 40 iterations. This indicates that our distributed approach can achieve the same accuracy as the centralized one, thus preserving privacy at no cost for accuracy.

As we have shown in Fig. 4 and Fig. 5, the model computed from the centralized approach is the same as that from our distributed approach. Therefore, we only need to compare the testing error rates of models computed by the distributed approach and the local approach. Since for the local approach, the performance depends on the size of training set for each user, we compare them under different user numbers N to see the influence of local data size on the error rates. Note that when $N = 1$, the performance of the two approaches are the same since both are identical to the centralized approach. We set $N = 100, 200, \ldots, 1000$ and then randomly divide the original datasets into smaller training sets, respectively. The testing error rates of our distributed approach and the local approach are shown in Fig. 6. On one hand, the error rate of the local approach increases as N increases (i.e., sample size per user decreases) due to the lack of diversity through data sharing. On the other hand, the error rate of our approach does not change w.r.t N because our algorithm always converges to the global optimal solution. This shows the advantage of collaborative learning by utilizing datasets sensed by multiple users.

The maximum computation time for any user at each iteration in our algorithm is 0.007 sec. The total time for the distributed approach to converge is around 0.12 sec. Therefore, our approach converges fast to the global optimal solution and incurs small computation overhead for each user.

5.2 Results on Imagery Task Classification (Vertically Partitioned Dataset)

In this section, we demonstrate the performance of our distributed approach for vertically partitioned data (Algorithm 2). Since no vertically partitioned medical database is readily available, we utilize dataset I of the Brain-Computer Interface Competition III (BCI-III-I) [4], [36] to simulate our scenario. The dataset records electrocorticography (ECoG) data of epileptic patients. In the original setting, each patient is sensed by 64 implanted electrodes covering certain locations of the cortex. However, here we assume that data for each trial is collaboratively sensed by 64 patients with 1 electrode implanted in each patient. A total of 278 trials are performed for data collection. Each trial starts with a cue of an imagery task (tongue or finger movement), and patients are required to mentally follow the cue. Their ECoG data during a 3-second imagination phase are sampled at sampling rate of 10Hz, resulting in 30×64 data points or a 1920-dimensional feature vector per trial. The data for all 278 trials form a 278×1920 matrix. Each row of the matrix represents data sampled with the same imagery cue by all 64 patients and has the same label ($+1$ or -1 for two different imagery tasks, respectively). The number of available training points is relatively small compared to the dimensionality of the data signal, which is a common case for vertically partitioned databases and can be effectively addressed by our distributed approach. Note that we use data sensed by a single patient to simulate data sensed by 64 patients, thus the training result may deviate from the original vertically partitioned setting. However, the goal of this experiment is to test the performance of our distributed approach rather than obtaining the regression parameters, thus this deviation does not invalidate our conclusion. In each experimental trial, we randomly select 200 training samples (100 positive, 100 negative) and 78 testing samples (36 positive, 36 negative) with each sample collaboratively sensed by 64 users. The testing error rates are the average results of 10 experimental trials.

Once again, we observe good convergence properties of our algorithms for different numbers of Patients N and therefore, for simplicity, we only show the convergence results of our distributed algorithm for $N = 64$. Fig. 7 illustrates the change of the logistic regression parameters w.r.t. the iteration number k when the dataset is vertically partitioned into $N = 64$ subsets. The figure shows that the difference between the regression parameters obtained by our algorithm and the optimal parameters converges to zero within tens of iterations. Fig. 7 shows the change of the

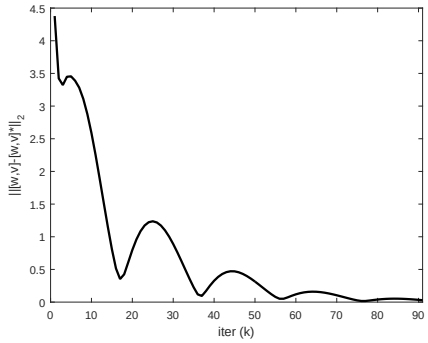


Fig. 7. Convergence of the logistic regression parameters obtained from our distributed approach on the vertically partitioned dataset.

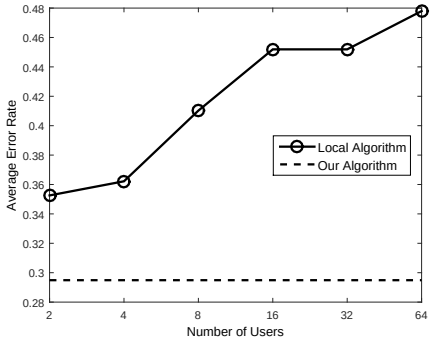


Fig. 9. Testing errors for our distributed approach and the local approach on the vertically partitioned dataset.

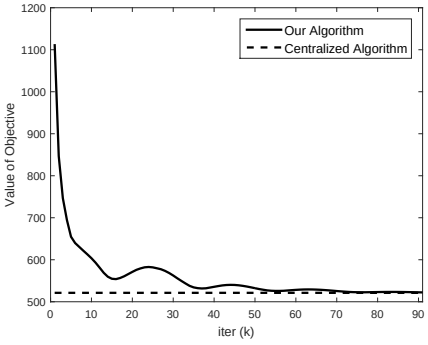


Fig. 8. Comparison of the objective of our approach (solid line) and the optimal objective (dashed line) on the vertically partitioned dataset.

objective function value w.r.t. iteration number. The solid line indicates the objective value obtained by the distributed approach, and the dashed line denotes the optimal objective value obtained by the centralized approach. As shown in the figure, the objective value of our approach decreases fast in the first few iterations and finally approaches the global minimum after 50 iterations. Therefore, our distributed approach can achieve the same accuracy as the centralized one.

Next, we compare the error rates of models computed by the distributed approach and the local approach. Since for the local approach, the performance depends on the size of training set for each user, we compare these two approaches under different user numbers N to see the influence of local data size on the error rates. When $N = 1$, the performance of the two approaches are the same since both are identical to the centralized approach. We set $N = 2, 4, 8, 6, 32, 64$ and randomly partition the original datasets into smaller training sets, respectively. Fig. 9 shows the error rates of our distributed approach and the local approach. We can see that the error rate of the local approach increases as N increases (i.e., sample size per user decreases) due to the lack of diversity through data sharing. The performance of our distributed approach does not depend on N since it always converges to the optimal solution after tens of iterations. The figure shows the benefit of data sharing in the vertically partitioned data.

The maximum computation time for any user at each iteration in our algorithm is 0.023 sec and the total time spent until convergence is 1.15 sec. Therefore, our approach is highly efficient even with 1960-dimensional data.

6 Related Work

There are a number of papers on private computation on medical data, but most of them focus on simple computations such as searching on encrypted medical data [37], computing statistical functions such as sum and variance [38], or performing predictive analysis tasks on encrypted data [26]. Few papers consider private model learning based on large-scale medical data despite of its great potential for healthcare quality and efficiency improvements. There are, however, several approaches for private model learning in general as summarized below.

Anonymization: One of the most popular ways for privacy-preserving learning is to anonymize the data by hiding the identity of the data source [39], [40]. However, it is possible to re-identify the data source. Narayanan and Shmatikov design a linkage attack that identifies personal information by linking two or more separate datasets [41]. A recent study in medical data demonstrates that individuals with detailed medical profiles are re-identified among anonymized medical data [42].

Perturbation-Based Approach: Another approach is to perturb the data content before transmitting it to the centralized party [43]–[46]. Fong and Weber-Jahnke [43] transform the original training samples into unreal data samples and use the unreal data samples for decision tree learning. However, perturbation always introduces error in the modeling process, trading accuracy for privacy. A modern privacy definition related with this approach is differential privacy, which requires that the output of a computation be equally likely with or without an input record [31]. The most common way to achieve differential privacy is through adding random noise [32]. In [47], McSherry and Mironov design a privacy-preserving scheme for training a recommendation system by adding differentially private noise to user data. Our approach is orthogonal to differential privacy due to differences in threat models. Differential privacy protect private information contained in the final computational results by injecting noise to the results, while we aim to protect private information during the computation process such that the party who performs the computation learns nothing more than the computational results.

Secure Multi-Party Computation: Secure multi-party computation-based approach is a conventional approach to training classifiers based on private data owned by multiple parties. A combination of cryptographic techniques is used to compute a function of their private data [48]–[50]. This approach usually guarantees that none parties can learn anything beyond what is contained in the final result. However, the cryptographic techniques used

in secure multi-party computation usually incur high computation cost, which is impractical for mHealth applications due to limited computing resources of mobile devices.

Homomorphic Encryption: Gentry [51] provides a fully homomorphic encryption solution for privacy-preserving computation, which avoids the need for two non-colluding parties. However, logistic regression involves a large number of both multiplication and addition steps. In this situation, current solutions for fully homomorphic encryption are not quite efficient [52], [53]. Although Lauter et al. [53] mention that their fully homomorphic encryption scheme can be used for regression, they do not show its performance. Graepel et al. train encrypted classifiers on encrypted training data using leveled homomorphic encryption [52], however, the efficiency of their approaches degrades rapidly when the size of the training data increases.

For model learning based on large-scale biomedical sensing data, it is important that the training algorithms scale well as the number of Patients increases. Most of the aforementioned cryptographic solutions incur high computation or communication load at the Patient side, and thus cannot be directly applied to our scenario. We address the scalability problem by decomposing the centralized optimization problem into subproblems such that the computation cost per Patient does not greatly increase with the number of Patients. Specifically, the decomposition algorithm we use is based on ADMM, which has been previously used for decomposing support vector machine (SVM) in [54]. Due to the decomposition, only the average of locally optimal parameters are needed by the mHealth server. Thus we can utilize a simple secure summation protocol with low amortized computational cost to protect private intermediate results. This paper is an extension of its conference version [55], with a new solution for vertical-partitioned healthcare data, more in-depth explanations of our approach, and a more extensive experimental evaluation.

7 Conclusion

In this paper, we have proposed a private scheme for learning a logistic regression model based on distributed biomedical sensing data. Our scheme enables mHealth users to control their raw data and only share necessary intermediate results during the training process. We have further provided a solution to protecting the private information of intermediate results during the aggregation process. Experimental results on real-world datasets show that the proposed approaches converge quickly and provide performance closely to the optimal result. Our schemes have low computational overhead for each user even when the number of users is large, and are thus practical for mHealth monitoring scenarios. We have focused on the logistic regression problem in this paper. However, our scheme may be generalized to other classification problems (e.g., support-vector machine) in mHealth applications which constitutes our future work.

References

[1] P. Mohan, D. Marin, S. Sultan, and A. Deen, "Medinet: personalizing the self-care process for patients with diabetes and cardiovascular disease using mobile telephony," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 755–758.

[2] H. Lin, J. Shao, C. Zhang, and Y. Fang, "Cam: cloud-assisted privacy preserving mobile health monitoring," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 6, pp. 985–997, 2013.

[3] R. Etherington, "Biostamp temporary tattoo electronic circuits by mc10," *Last Update: March 28th*, 2013.

[4] J. d. R. Millán, "On the need for on-line learning in brain-computer interfaces," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 4. IEEE, 2004, pp. 2877–2882.

[5] L. Hood and N. Price, "Promoting wellness & demystifying disease: the 100k project," *Clinical OMICs Innovator, May 2014*, 2014.

[6] 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.

[7] R. B. D'Agostino, M. J. Pencina, J. M. Massaro, and S. Coady, "Cardiovascular disease risk assessment: Insights from framingham," *Global heart*, vol. 8, no. 1, pp. 11–23, 2013.

[8] 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.

[9] S. Jiménez-Serrano, S. Tortajada, and J. M. García-Gómez, "A mobile health application to predict postpartum depression based on machine learning," *Telemedicine and e-Health*, 2015.

[10] C. R. Boyd, M. A. Tolson, and W. S. Copes, "Evaluating trauma care: the triss method." *Journal of Trauma and Acute Care Surgery*, vol. 27, no. 4, pp. 370–378, 1987.

[11] R. Blankstein, R. P. Ward, M. Arnsdorf, B. Jones, Y.-B. Lou, and M. Pine, "Female gender is an independent predictor of operative mortality after coronary artery bypass graft surgery contemporary analysis of 31 midwestern hospitals," *Circulation*, vol. 112, no. 9 suppl, pp. I–323, 2005.

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

[13] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 5–20.

[14] X. Bao and R. Roy Choudhury, "Movi: mobile phone based video highlights via collaborative sensing," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 357–370.

[15] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han, "Privacy-aware regression modeling of participatory sensing data," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 99–112.

[16] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & Sons, 2005, vol. 589.

[17] 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.

[18] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Advances in Neural Information Processing Systems*, 2009, pp. 289–296.

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

[20] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, 1975.

[21] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[22] 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.

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

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

[25] A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter, "Privacy preserving regression modelling via distributed computation," in *KDD*, 2004.

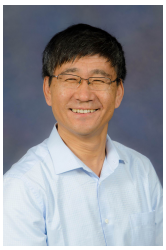
[26] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data," *Journal of biomedical informatics*, vol. 50, pp. 234–243, 2014.

[27] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM Sigkdd Explorations Newsletter*, vol. 4, no. 2, pp. 28–34, 2002.

[28] 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.

[29] D. J. Bernstein and T. Lange, "ebacs: Ecrypt benchmarking of cryptographic systems," 2009.
[30] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology-EUROCRYPT 2006*. Springer, 2006, pp. 486–503.
[31] C. Dwork, "Differential privacy," in *Proceedings of the 33rd international conference on Automata, Languages and Programming-Volume Part II*. Springer-Verlag, 2006, pp. 1–12.
[32] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Theory of Cryptography Conference*, 2006, pp. 265–284.
[33] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*. IEEE, 2007, pp. 94–103.
[34] Y. Duan, "Privacy without noise," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1517–1520.
[35] PDMC Workshop, 2004. [Online]. Available: http://algoval.essex.ac.uk/data/series/pdmc
[36] BCI Competition III. [Online]. Available: http://www.bbci.de/competition/iii/
[37] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 103–114.
[38] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 2011, pp. 113–124.
[39] 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.
[40] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 238–243.
[41] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 111–125.
[42] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.
[43] 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.
[44] K.-P. Lin and M.-S. Chen, "On the design and analysis of the privacy-preserving svm classifier," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 11, pp. 1704–1717, 2011.
[45] A. Evfimievski, "Randomization in privacy preserving data mining," *ACM Sigkdd Explorations Newsletter*, vol. 4, no. 2, pp. 43–48, 2002.
[46] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 439–450.
[47] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the net," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 627–636.
[48] J. Vaidya and C. Clifton, "Privacy-preserving decision trees over vertically partitioned data," in *Data and Applications Security XIX*. Springer, 2005, pp. 139–152.
[49] J. Vaidya, M. Kantarcıoğlu, and C. Clifton, "Privacy-preserving naive bayes classification," *The VLDB JournalłThe International Journal on Very Large Data Bases*, vol. 17, no. 4, pp. 879–898, 2008.
[50] J. Vaidya, H. Yu, and X. Jiang, "Privacy-preserving svm classification," *Knowledge and Information Systems*, vol. 14, no. 2, pp. 161–178, 2008.
[51] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.
[52] 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.
[53] K. Lauter, M. Naehrig, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 2011, pp. 113–124.
[54] K. Xu, H. Yue, L. Guo, Y. Guo, and Y. Fang, "Privacy-preserving machine learning algorithms for big data systems," in *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*. IEEE, 2015, pp. 318–327.
[55] Y. Gong, Y. Fang, and Y. Guo, "Privacy-preserving collaborative learning for mobile health monitoring," in *IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, December 6-10 2015.

Yanmin Gong (S'10) received the B.Eng. degree in electronics and information engineering from Huazhong University of Science and Technology, China, and the M.S. degree in electrical engineering from Tsinghua University, China, in 2009 and 2012, respectively. She is currently a PhD student in electrical and computer engineering at the University of Florida. Her research interest includes security and privacy in big data with emphasis on healthcare, mobile systems, and energy.



Yuguang Fang (F'08) received MS degree from Qufu Normal University, China, in 1987, and Ph.D. degrees from both Case Western Reserve University and Boston University in 1994 and 1997, respectively. He joined the Department of Electrical and Computer Engineering at University of Florida since 2000. Dr. Fang received the US NSF Faculty Early Career Award in 2001 and the US ONR Young Investigator Award in 2002, and is a recipient of the Best Paper Award in IEEE International Conference on Network Protocols in 2006. He also received a 2010-2011 UF Doctoral Dissertation Advisor/Mentoring Award and IEEE Communications Society WTC Recognition Award. He served as the Editor-in-Chief of IEEE Wireless Communications and is currently serving as the Editor-in-Chief of IEEE Transactions on Vehicular Technology. He is a Fellow of IEEE.

Yuanxiong Guo (M'14) received the B.Eng. degree in electronics and information engineering from Huazhong University of Science and Technology, China, in 2009 and the M.S. degree and Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012 and 2014, respectively.

He is now an Assistant Professor in the School of Electrical and Computer Engineering at Oklahoma State University, Stillwater, OK, USA. His research interests include smart grids, cyber-physical systems, sustainable computing and networking, and critical infrastructure cybersecurity and resilience. He is the recipient of the Best Paper Award in IEEE Global Communications Conference 2011.