Secure and Differentially Private Logistic Regression for Horizontally Distributed Data

Miran Kim, Junghye Lee, Lucila Ohno-Machado, and Xiaoqian Jiang

***Abstract*— Scientific collaborations benefit from sharing information and data from distributed sources, but protecting privacy is a major concern. Researchers, funders, and the public in general are getting increasingly worried about the potential leakage of private data. Advanced security methods have been developed to protect the storage and computation of sensitive data in a distributed setting. However, they do not protect against information leakage from the outcomes of data analyses. To address this aspect, studies on differential privacy (a state-of-the-art privacy protection framework) demonstrated encouraging results, but most of them do not apply to distributed scenarios. Combining security and privacy methodologies is a natural way to tackle the problem, but naive solutions may lead to poor analytical performance. In this paper, we introduce a novel strategy that combines differential privacy methods and homomorphic encryption techniques to achieve the best of both worlds. Using logistic regression (a popular model in biomedicine), we demonstrated the practicability of building secure and privacy-preserving models with high efficiency (less than 3 min) and good accuracy [<1% of difference in the area under the receiver operating characteristic curve (AUC) against the global model] using a few real-world datasets.**

***Index Terms*— Logistic regression, differential privacy, homomorphic encryption.**

I. Introduction

BIOMEDICINE collaborations between biologists and researchers often require data sharing. For example, researchers might want to learn the impact of certain variants for some rare diseases, but an institution may not have enough samples to support the intended data analysis. They want to collaborate, but due to institutional policies or other reasons, it may not be feasible to move data from one institution to another.

Big data networks, such as Electronic Medical Records and Genomics (eMERGE) Network [1], the patient-centered SCAlable National Network for Effectiveness Research (pSCANNER) clinical data research network (CDRN) [2], the Scalable Architecture for Federated Translational Inquiries Network (SAFTINet) [3] have been established to accelerate discovery, promote personalized medicine, and improve public health. However, biomedical data integration and sharing raise public concerns that information exchange (e.g. demographics, diagnostic codes, medication, genome sequences) or distributed data analysis can put sensitive patient information at risk. A breach may have serious implications for research participants such as discrimination for employment, for certain types of insurance, or even for education [4].

Differential Privacy (DP) has emerged as one of the strongest privacy guarantees for statistical information release [5]. Many recent studies developed differentially private algorithms for interactive query answering, histogram and synthetic data publication [6]–[10]. Most of these methods deal with centralized datasets. In reality, data are often distributed geographically across institutions. In addition, there does not always exist a trusted authority in ad-hoc cross-institutional collaborations. To promote efficient yet privacy-preserving collaboration, it is imperative to develop methods that "bring computation to data", rather than "bringing data to the computation". In fact, we need to consider security as well as privacy in distributed data analyses. Note that privacy and security are different: the former ensures the output of data analysis does not leak information about a private dataset while the latter guarantees the confidentiality during information exchange or "at rest". These two objectives do not always work well together and naive solutions might suffer from low efficiency and poor performance. An alternative cryptographic primitive for secure computation is *Homomorphic Encryption* (HE), which allows to perform arithmetic operations on encrypted data without the need for decryption. It is ideal for secure outsourcing but it needs to be carefully designed to avoid execution of deep circuits, otherwise the computation can be formidable.

A. Our Contributions

We focus on a *horizontally partitioned* setting, where multiple institutions holding sensitive data of sub-population

Manuscript received July 9, 2018; revised November 23, 2018, March 27, 2019, and May 30, 2019; accepted June 5, 2019. Date of publication June 27, 2019; date of current version September 24, 2019. The work of M. Kim and X. Jiang was supported in part by the Cancer Prevention Research Institute of Texas (CPRIT) under Award RR180012 and in part by the National Institute of Health (NIH) under Award R01GM118609 and Award U01EB023685. The work of J. Lee was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2018R1C1B5086611. The work of L. Ohno-Machado was supported by the NIH under Award R01GM118609 and Award R01HL136835. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aris Gkoulalas Divanis. *(Corresponding author: Junghye Lee.)*

M. Kim and X. Jiang are with the School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX 77030 USA (e-mail: miran.kim@uth.tmc.edu; xiaoqian.jiang@uth.tmc.edu).

J. Lee is with the School of Management Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, South Korea (e-mail: junghyelee@unist.ac.kr).

L. Ohno-Machado is with the Health Department of Biomedical Informatics, University of California at San Diego, San Diego, CA 92093 USA (e-mail: lohnomachado@ucsd.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2019.2925496

want to collaboratively learn a model from their data without compromising their privacy. The major contributions of this work can be summarized as follows:

- We present a hybrid approach to conduct distributed logistic regression in a secure and privacy-preserving manner, which combines DP and HE. We explore an iterative learning algorithm with objective perturbation, where a semi-honest server learns a differential private model from homomorphically encrypted local summary statistics at each iteration. Therefore, our approach allows privacy protection of both data and analysis outcomes during the learning process.
- We propose two protocols that are based on different diagonal approximations of the Hessian matrix in the Newton-Raphson method. Firstly, we adapt the fixed diagonal Hessian approximation, which is securely pre-computed in a distributed manner across institutions. Therefore, a server securely updates a model estimator from locally perturbed encrypted gradients at each iteration. The updated estimator is decrypted by the secret-key owner and re-distributed to local institutions for the next iteration. Secondly, we explore the diagonal updating approach via quasi-Cauchy relation in a distributed learning setting. At each iteration, local institutions need to compute additional summary statistics to ensure positive definiteness of the diagonally approximated Hessian matrix and send their encryptions to the server.
- We present theoretical computational complexity of the proposed methods and conduct extensive empirical evaluation to show their scalability on real data. The results show that our methods can produce secure and privacy-preserving predictive models with high efficiency and good accuracy.

II. Related Work

Chaudhuri and Monteleoni [11] proposed differentially private logistic regression models under two different mechanisms: (1) output perturbation that introduces noises on the learned parameters, and (2) objective perturbation that introduces an additional noisy term on the objective function to ensure the final results are differentially private. The following studies [12], [13] are a generalized version of the objective perturbation in [11] for regularized empirical risk minimization algorithms. In particular, the latter was developed for improving the objective perturbation technique to use a smaller noise than the former and to be compatible with hard constraints and non-differentiable regularizers. Awan and A. Slavković [14] extended their objective perturbation mechanisms to K-norm mechanisms while optimizing the finite-sample performance. These models work under a centralized setting in which the private dataset is hosted and trained. Yu *et al.* [15] studied the penalty parameter selection problem using the objective perturbation method to provide an end-to-end differentially private logistic regression solution for detecting multiple-SNP association in GWAS databases. Ji *et al.* [16] developed a differentially private Newton-Raphson algorithm to optimize the logistic regression model that only adds noise to gradient (it uses the Hessian matrix obtained from a public dataset). The improvement of this method relies on the existence and availability of a public dataset that has similar distribution with the private dataset. Furthermore, both of above-mentioned methods are also developed for a centralized setting, and thus they are not directly applicable to distributed modeling. Recently, Hegedus and Jelasity [17] performed an empirical study of using distributed differentially private stochastic gradient descent (SGD) to train a logistic regression model. Unfortunately, the model requires a large number of iterations to converge and there is a noticeable gap between the accuracy of SGD and its non-differentially private counterpart (even when a large privacy budget is allocated).

A different approach makes use of *multi-party computation* (MPC), where two or more parties jointly compute a function on their inputs without revealing their data. Previous studies [18]–[22] showed the feasibility of building a secure distributed logistic regression across multiple institutions, but their methods suffer from scalability due to the complexity of the underlying secure computation primitives. Mohassel and Zhang proposed SecureML [23], a two-party secure framework for scalable privacy-preserving machine learning based on a server-aided secure MPC framework. Their work utilizes secret sharing protocols, which require collaborators to distribute their inputs to servers that will not collude, in order to develop efficient approximation algorithms. The computation of MPC is interactive and the communication cost grows quickly as the sample size and/or feature dimensionality increase(s). Recently, Snoke *et al.* [24] developed a method to perform maximum likelihood estimation of the parameters in a distributed setting. These approaches can guarantee the privacy of training data during the learning process, but not of the estimated model.

A possible solution is to securely exchange intermediate statistics of distributed logistic regression models and add perturbations when necessary. Several prior studies have proposed to combine DP and MPC for distributed learning. Pathak *et al.* [25] proposed a protocol that securely aggregates locally trained models using MPC and uses output perturbation to achieve different privacy. However, their model is less accurate due to the noise that is added before aggregations of local models. Shokri and Shmatikov [26] proposed an iterative updating method where the local gradients are perturbed and revealed before the update. Recently, Jayaraman *et al.* [27] improved on the best previous noise bounds of [25], [26] by adding noise directly to the aggregated model parameters or the aggregated local gradients in the MPC protocol. However, their approach has an inherent challenge such that the secret-sharing MPC model can be vulnerable when parties collude. In addition, the iterative learning scenario requires one MPC protocol execution at each iteration, resulting in high bandwidth usage.

Data analysis based on HE, which is another secure computation mechanism, is a new area of research but some recent solutions are still using hybrid approaches with other cryptographic primitive (e.g. MPC) or leveraging interactive mechanisms between data owners and the server.

Xie *et al.* [28] proposed a secure protocol to conduct a logistic regression in a distributed manner. Their method is based on Yao's garbled circuit and an additive HE scheme, which still gives rise to a heavy computation in computing some intermediate statistics. Phong *et al.* [29] used an additive property of HE to securely aggregate locally encrypted gradients, but the updated parameters are decrypted by the secret-key owner and their model cannot guarantee the privacy of the parameters. Recently, HE-based secure outsourcing solutions have demonstrated performance improvement in practical use of machine learning; for example, Kim *et al.* [30], [31] presented secure outsourcing methods to train a logistic regression model on encrypted data and showed their feasibility with real data.

A similar study to our approach was done by Aono *et al.* [32]. Their protocol securely aggregates the local information by using an additive HE scheme (securely computing a polynomial approximation of the objective function) and achieves DP using output perturbation. Their approach is computationally efficient; however, it can suffer from accuracy degradation and scalability in high-dimensional data.

III. Preliminaries

A. Logistic Regression

Logistic regression is very popular in biomedical informatics research and serves as the foundation of many risk calculators [33]–[35]. The logistic function for binary-valued outcomes can be written as follows: $\Pr(Y = y|z, \beta) = 1/(1 + \exp(-y\beta^T z))$ where y and z represent the observed outcome and the covariates (d-dimensional) of a sample respectively, and each dimension corresponds to an attribute. The parameters β measure the relationship between Y (response variable) and the covariates Z, and they are the target of the estimation. We can add a regularization term $\frac{\lambda}{2}\beta^T\beta$ to make the logistic regression objective a strong convex function. The parameter λ balances the bias and variance to avoid over-fitting, and it is commonly used in penalized logistic regression [36] as follows:

$$l(\beta) = -\sum_{i=1}^{n} \log(1 + \exp(-y_i \beta^T z_i)) - \frac{\lambda}{2}\beta^T\beta, \tag{1}$$

where n is the number of samples, z_i and y_i stand for the i-th sample and its binary class label, respectively. There is no closed-form solution for β which maximizes the objective, so the estimation requires iterative numerical methods to obtain the final parameters using the maximum likelihood estimation [37]. We can use the Newton-Raphson algorithm to estimate parameters β, which can be achieved by calculating the first and second derivatives of the log-likelihood function:

$$\beta^{t+1} = \beta^t - (l''(\beta^t))^{-1} l'(\beta^t) \tag{2}$$

where t stands for the iteration. If data are centralized, the derivatives can be calculated as follows:

$$l''(\beta^t) = -Z^T R^t Z - \lambda I, \quad l'(\beta^t) = Z^T(y - \mu^t) - \lambda\beta^t, \tag{3}$$

where R^t is an $n \times n$ diagonal matrix of weights with the ith element $\Pr(Y = 1|z_i, \beta^t) \cdot (1 - \Pr(Y = 1|z_i, \beta^t))$, and μ^t is an n-dimensional vector with the i-th element $\Pr(Y = 1|z_i, \beta^t)$.

When data are distributed, necessary intermediary statistics (i.e., Hessian and gradient) for maximum likelihood estimation of logistic regression model can be linearly decomposed and locally calculated at individual sites:

$$l''(\beta^t) = -\sum_k (Z_k^T R_k^t Z_k + \frac{\lambda}{K} I),$$
$$l'(\beta^t) = \sum_k (Z_k^T (y_k - \mu_k^t) - \frac{\lambda}{K}\beta^t), \tag{4}$$

where $k \in [1, K]$ represents different collaborative sites [38].

B. Differential Privacy

DP has emerged as one of the strongest privacy-preserving solutions for aggregating sensitive data.

Definition 1 (ϵ-DP [39]): A randomized algorithm Ag *satisfies ϵ-DP if for all data sets D and D' where their symmetric difference contains at most one record (i.e., $|D \Delta D'| \leq 1$), and for all $\mathcal{S} \subseteq Range(\mathsf{Ag})$, we have* $\Pr[\mathsf{Ag}(D) \in \mathcal{S}] \leq e^{\epsilon} \cdot \Pr[\mathsf{Ag}(D') \in \mathcal{S}]$, *where the probabilities are over the randomness of the algorithm* Ag. *The parameter ϵ is called the privacy budget, for which a small budget corresponds to stronger protection and vice versa.*

Definition 2 (Sensitivity [39]): For any function $f : D \to \mathcal{R}^d$, the sensitivity of f is defined as $\Delta f = \max_{D,D'} \|f(D) - f(D')\|_1$, for all D and D' differing by at most one record.

Definition 3 (Laplace Mechanism [40]): The mechanism takes as inputs a data set D, a function $f : D \to \mathcal{R}^d$, and the parameter $\Delta f/\epsilon$ that determines the magnitude of noise. It first computes the true output $f(D)$, and then perturbs the output by adding noise as $f(D') = f(D) + Lap(0, \Delta f/\epsilon)$. The mechanism that adds independently generated noise L with the distribution $Lap(0, \Delta f/\epsilon)$ to each of the d output terms enjoys ϵ-DP.

In a distributed setting, DP can be achieved by Distributed Laplace Perturbation Algorithms (DLPAs). In DLPA, each party generates a partial noise where the aggregated noise follows the Laplace distribution (i.e., $L \sim Lap(0, \alpha)$), which is enough to guarantee DP. Due to infinite divisibility of the Laplace distribution [41], a random variable (r.v.) L with such distribution can be computed by summing up K other r.v.s.. We utilize this property and explore a few algorithms named after different distributions to draw such partial noise (i.e., Gaussian, Gamma, and Laplace), respectively.

- Gaussian DLPA [42]: L can be simulated by four independent and identically distributed (i.i.d.) r.v.s N_1, N_2, N_3, N_4, and each is drawn from the normal distribution $N(0, \alpha/2)$ and applied as follows, $L = N_1^2 + N_2^2 - N_3^2 - N_4^2$. Drawing a single r.v. $N_i \sim N(0, \alpha/2), i = 1, ..., 4$, is again simulated by the sum of K i.i.d. Gaussian r.v.s $N_{i,k} \sim N(0, \alpha/2K)$, i.e., $N_i = \sum_{k=1}^{K} N_{i,k}$.
- Gamma DLPA [43]: L can be generated by the sum of $2K$ r.v.s, $L = \sum_{k=1}^{K} (G_k - H_k)$, where G_k and H_k are i.i.d. gamma distributed r.v.s with the parameters $1/\alpha$ and $1/K$.

- Laplace DLPA [44]: L can be drawn by K i.i.d. r.v.s $L_k \sim Lap(0, \alpha)$ and a single r.v. $B \sim Beta(1, K-1)$, as defined in $L = \sqrt{B} \cdot \sum_{k=1}^{K} L_k$.

C. Approximate Homomorphic Encryption

HE is a cryptographic tool that allows computation on encrypted data without the need for decryption and generates an encrypted result matching that of operations on plaintext. So, it enables us to securely outsource expensive computation on an untrusted server. This technology has great potentials in many real-world applications such as statistical testing, neural networks, and machine learning [30], [31], [45]–[47].

In this paper, we employ a special cryptosystem [48] with support for *approximate arithmetic* of encrypted messages, called HE for Arithmetic of Approximate Numbers (HEAAN). The main idea is to consider an encryption noise (on the ciphertext for security) as a part of computation error that occurs in approximate arithmetic. That is, given a secret key sk, an encryption ct of a plaintext m satisfies the equation $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = m + e \pmod{q}$ for some small e. It supports an encryption function $\mathsf{Enc}(\cdot)$ and a decryption function $\mathsf{Dec}(\cdot)$ such that

$$\mathsf{Dec}(\mathsf{Enc}(x + y)) \approx \mathsf{Dec}(\mathsf{Enc}(x)) + \mathsf{Dec}(\mathsf{Enc}(y)),$$
$$\mathsf{Dec}(\mathsf{Enc}(x \cdot y)) \approx \mathsf{Dec}(\mathsf{Enc}(x)) \cdot \mathsf{Dec}(\mathsf{Enc}(y)).$$

This HE scheme provides a trade-off between precision and efficiency but it offers a practical and effective solution for some applications that do not require absolute precision.

Let N be a power-of-two integer and $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ be the ring of integers of the $(2N)$-th cyclotomic field. Let us denote by $[\cdot]_q$ the reduction modulo q into the interval $(-q/2, q/2] \cap \mathbb{Z}$. We write $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ for the residue ring of $\mathcal{R}$ modulo an integer q. The following is a simple description of the HEAAN scheme based on the Ring Learning with Errors (RLWE) problem.

- $\underline{\mathsf{ParamsGen}(\lambda)}$. Given the security parameter λ, choose a power-of-two integer N, a modulus $Q = q^2$, and a discrete Gaussian distribution χ. The RLWE problem of parameter (N, Q, χ) should achieve at least λ bits of security level for the semantic security of cryptosystem. Output $params \leftarrow (N, q, \chi)$.
- $\underline{\mathsf{KeyGen}(params)}$. Generate $s \in \mathcal{R}$ by sampling its coefficient randomly from $\{0, \pm 1\}^N$ and set the secret key as $\mathsf{sk} \leftarrow (1, s)$. Sample a uniformly at random from $\mathcal{R}_q$ and e from χ. Set the public key as $\mathsf{pk} \leftarrow (b, a) \in \mathcal{R}_q \times \mathcal{R}_q$ where $b \leftarrow -as + e \pmod{q}$. Let $s' \leftarrow s^2$. Sample a' uniformly at random from $\mathcal{R}_Q$ and e' from χ. Set the evaluation key as $\mathsf{evk} \leftarrow (b', a') \in \mathcal{R}_Q \times \mathcal{R}_Q$ where $b' \leftarrow -a's + e' + qs' \pmod{Q}$.
- $\underline{\mathsf{Enc}(m, \mathsf{pk})}$. For $m \in \mathcal{R}_q$, sample a small polynomial v (with $0, \pm 1$ coefficients) and two error polynomials e_0, e_1 from χ. Output $v \cdot \mathsf{pk} + (m + e_0, e_1) \in \mathcal{R}_q \times \mathcal{R}_q$.
- $\underline{\mathsf{Dec}(\mathsf{ct}, \mathsf{sk})}$. For $\mathsf{ct} = (c_0, c_1) \in \mathcal{R}_{q'}^2$, output $m \leftarrow c_0 + c_1 \cdot s \pmod{q'}$.
- $\underline{\mathsf{Add}(\mathsf{ct}, \mathsf{ct}')}$. For $\mathsf{ct}, \mathsf{ct}' \in \mathcal{R}_{q'}^2$, output $\mathsf{ct}_{\mathsf{add}} \leftarrow \mathsf{ct} + \mathsf{ct}' \pmod{q'}$.
- $\underline{\mathsf{Mult}_{\mathsf{evk}}(\mathsf{ct}, \mathsf{ct}')}$. For two ciphertexts $\mathsf{ct} = (c_0, c_1), \mathsf{ct}' = (c_0', c_1') \in \mathcal{R}_{q'}^2$, let $(d_0, d_1, d_2) = (c_0c_0', c_0c_1' + c_1c_0', c_1c_1') \pmod{q'}$. Output $\mathsf{ct}_{\mathsf{mult}} \leftarrow (d_0, d_1) + \lfloor q^{-1}d_2 \cdot \mathsf{evk} \rceil \pmod{q'}$.
- $\underline{\mathsf{RS}(\mathsf{ct}; r)}$. For $\mathsf{ct} \in \mathcal{R}_{q'}^2$, output $\mathsf{ct}_{\mathsf{rs}} \leftarrow \lfloor r^{-1} \cdot \mathsf{ct} \rceil \pmod{r^{-1}q'}$.

For a power-of-two integer $\ell \leq N/2$, HEAAN provides a technique to pack ℓ complex numbers in a single polynomial using a variant of the complex canonical embedding map $\phi : \mathbb{C}^\ell \rightarrow \mathcal{R}$. Throughout this paper, we restrict the plaintext space as a vector of real numbers. We multiply a scale factor of p to plaintexts before the rounding operation in order to preserve their precision during computation. Since addition and multiplication in $\mathcal{R}$ correspond the component-wise addition and multiplication on the plaintext vector, we can parallelize the operations and alleviate the costs of space and computation time. We refer the reader to [48] for the technical details and noise analysis.

IV. Secure and Differentially Private Distributed Logistic Regression

We aim to compute logistic regression for horizontally distributed data in a secure and privacy preserving manner. A straightforward idea is to apply distributed DP on the intermediate statistics of distributed logistic regression model (introduced in Section III-B). However, if we directly combine these methods, the noise added by any single party is not sufficient to ensure the expected global DP. In this section, we introduce a novel solution that synergistically harmonizes DP and HE into a win-win strategy to develop secure and differentially private distributed logistic regression. At a high level, our protocol protects the intermediate statistics and their computation during each iteration of parameter updates while decrypting their differentially private intermediate model estimator in order to avoid a heavy computational cost (due to accumulation of deep circuits in HE). We note that it strictly satisfies the differential privacy criterion, which is described later in Section IV-C.

There are three parties in our protocol: collaborative sites $P_1, \ldots, P_K$ (data providers), a cryptographic service provider (CSP), and a cloud server. The CSP manages the cryptographic keys that are used for encryption (pk), decryption (sk), and homomorphic computation (evk). The cloud server is where encrypted data is processed while it is still in encrypted form. Fig. 1 illustrates our protocol.

We consider the following threat models. Firstly, we assume that the cloud server is semi-honest, that is, an adversary is honest but curious. If we ensure the semantic security of the underlying HE scheme, all the computations on the server are performed over encryption so the server learns nothing from the encrypted data. Secondly, we assume that the CSP is not allowed to collude with the server or each site. The CSP should not be given access to data that are not part of the query from the server. Lastly, we assume that the local sites should not collude. That is, their local sample sizes are not disclosed to the other local parties even though the global sample size is given to all the local sites.

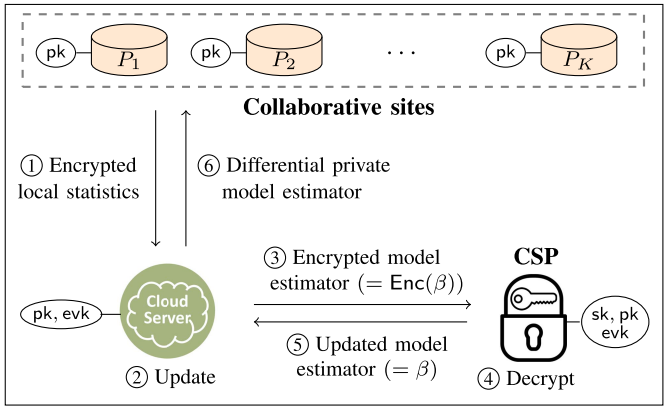


Fig. 1. Overview of our secure and privacy-preserving logistic regression protocol.

Throughout this paper, we employ the objective perturbation approach in [12]:

$$l(\beta) = -\sum_{i=1}^{n} \log(1 + \exp(-y_i \beta^T z_i)) - \frac{\lambda}{2}\beta^T\beta + \frac{b^T\beta}{n}, \tag{5}$$

$$l''(\beta^t) = -Z^T R^t Z - \lambda I = \sum_{k=1}^{K} \left(-Z_k^T R_k^t Z_k - \frac{\lambda}{K} I \right), \tag{6}$$

$$\begin{aligned} l'(\beta^t) &= Z^T(y - \mu^t) - \lambda\beta^t + \frac{b}{n} \\ &= \sum_{k=1}^{K} \left(Z_k^T (y_k - \mu_k^t - \frac{\lambda}{K}\beta^t) + \frac{b_k}{n} \right), \end{aligned} \tag{7}$$

where n is the global sample size, K is the total number of participating institutions, and b is the noise aggregated from locally generated noise b_k according to DLPAs.

The existing HE schemes do not naturally support matrix inversion, which makes it difficult to compute the inverse of the Hessian over encrypted data (as expected in Equation 2). Recently, Jiang *et al.* [47] developed a solution to encrypt a matrix into a single ciphertext and perform arithmetic operations using the ciphertext packing method. However, this method is not generalized to matrix inverse, so we come up with an alternative solution in this paper. The basic observation is that the inverse of a diagonal matrix is obtained by replacing each element in the diagonal with its reciprocal. We adapt the diagonal approximation of the Hessian matrix and apply the inverse method of [48] to our protocol. This approach has another advantage in that it significantly reduces the complexity of multiplication between the inverse of Hessian and the gradient by employing the single instruction multiple data (SIMD) technique. That is, we only need to perform component-wise multiplication between encrypted vectors. In the following, we introduce methodologies related to two kinds of diagonal Hessian approximations.

A. Fixed Diagonal Hessian Method

The Newton-Raphson algorithm can be accelerated by using an approximation of the Hessian to a fixed matrix $\bar{H}$, which only needs to be inverted once. For maximum likelihood estimation, Böhning [49] suggested the matrix $\bar{H} = -\frac{1}{4}Z^T Z$, which can be generalized to $\bar{H} = -\frac{1}{4}Z^T Z - \frac{\lambda}{K} I$. Here, we introduce a simple approximation of $\bar{H}$ using only its diagonal elements: $\bar{H} \approx \text{diag}(\bar{H}) = \tilde{H}$. The fixed diagonal Hessian approximation $\tilde{H}$ can be locally decomposed as $\sum_k \tilde{H}_k$ such that

$$\tilde{H}_k = \text{diag}\left(-\frac{1}{4} Z_k^T Z_k - \frac{\lambda}{K} I \right). \tag{8}$$

Because the fixed diagonal Hessian does not depend on the parameters β, we can pre-compute its inverse only once. However, this approach has a limitation that the result would only be valid when the Hessian matrix is strongly diagonal-dominant, which means it is largely dependent on the parameter λ to be set.

Our secure and differentially private distributed logistic regression using the fixed diagonal Hessian approximation is described in Algorithm 1, denoted by F-SPLR. It consists of two phases: a preparation phase of securely inverting the Hessian and determining the global sample size, and an iterative estimation phase of the model estimator β.

1) Preparation Phase: The CSP first generates the key trio $(\mathsf{sk}, \mathsf{pk}, \mathsf{evk})$ while each local institution only has access to the public key pk for data encryption and the cloud server can only access to the evaluation key evk for homomorphic computation (Step 1). Next, each local site generates a local random noise b_k from DLPAs (Step 3), which will be added to the local gradient. Then it computes the local diagonal Hessian approximation $\tilde{H}_k$ as in (8), encrypts the plaintext vector using pk, and sends the output ciphertext to the cloud server (Step 4). In addition, it encrypts the local sample size n_k and transmits $\mathsf{Enc}(n_k)$ to the server (Step 5).

The server securely aggregates the local diagonal Hessian across institutions and computes the inverse of the global diagonal Hessian (Step 7). As mentioned before, we adapt the multiplicative inverse evaluation algorithm, denoted by $\mathsf{Inv}(\cdot)$, suggested in [48]. In other words, we compute $\mathsf{Inv}(\sum_k \mathsf{Enc}(\tilde{H}_k))$ over encryption, or equivalently, we obtain a ciphertext that encrypts an approximated value to

$$\begin{aligned} \mathsf{Dec}(\mathsf{Inv}(\textstyle\sum_k \mathsf{Enc}(\tilde{H}_k))) &\approx \mathsf{Dec}(\mathsf{Inv}(\mathsf{Enc}(\tilde{H}))) \\ &\approx \mathsf{Dec}(\mathsf{Enc}(\tilde{H}^{-1})) \approx \tilde{H}^{-1} \end{aligned} \tag{9}$$

from homomorphic properties of the HEAAN scheme. The server also aggregates the local sample sizes $\mathsf{Enc}(n_k)$ (Step 8) and the resulting ciphertext is decrypted with the secret key of the CSP (Step 9). If we take a sufficiently large scale factor of p, the global sample size n can be the nearest integer of the decryption result $\mathsf{Dec}(\sum_k \mathsf{Enc}(n_k))$. Thus we get the desired sample size by computing $\lfloor \mathsf{Dec}(\sum_k \mathsf{Enc}(n_k)) \rceil = \lfloor \mathsf{Dec}(\mathsf{Enc}(n)) \rceil = n$, and CSP disseminates the global sample size to each local site.

2) Iterative Estimation Phase: At each iteration t, each site computes the locally perturbed gradient by

$$g_{tk} = Z_k^T (y_k - \mu_k^t - \frac{\lambda}{K}\beta^t) + \frac{b_k}{n}, \tag{10}$$

Algorithm 1 F-SPLR: homomorphic evaluation of the fixed diagonal Hessian method

Input: Initial $\beta^0 = 0$; regularization parameter λ; DP budget ϵ; number of iterations of regression ι
Output: Globally estimated coefficients β

[At the CSP]:
1: Generate keys: $(\mathsf{sk}, \mathsf{pk}, \mathsf{evk})$

[At local sites]:
2: **for** each site $k = 1$ to K **do**
3: Generate a local random noise b_k
4: Compute local fixed diagonal Hessian $\tilde{H}_k$, encrypt, and transmit $\mathsf{Enc}(\tilde{H}_k)$ to the server
5: Encrypt and transmit $\mathsf{Enc}(n_k)$ to the server
6: **end for**

[At the cloud server]:
7: Aggregate local fixed diagonal Hessians across sites and invert the global diagonal Hessian: $\mathsf{Inv}(\sum_k \mathsf{Enc}(\tilde{H}_k))$
8: Aggregate local sample sizes across sites and transmit to the CSP: $\sum_k \mathsf{Enc}(n_k)$

[At the CSP]:
9: Decrypt and disseminate the global sample size to each local site: $n \leftarrow \mathsf{Dec}(\sum_k \mathsf{Enc}(n_k))$

10: **for** $t = 0$ to $\iota - 1$ **do**
[At local sites]:
11: **for** each site $k = 1$ to K **do**
12: Compute local gradient g_k using the local noise b_k, encrypt, and transmit $\mathsf{Enc}(g_k)$ to the server
13: **end for**

[At the cloud server]:
14: Aggregate locally updated gradients: $\sum_k \mathsf{Enc}(g_k)$
15: Multiply the global gradient by the inverse Hessian: $\mathsf{ct}_{\mathsf{GD}} \leftarrow \mathsf{Inv}(\sum_k \mathsf{Enc}(\tilde{H}_k)) \cdot \sum_k \mathsf{Enc}(g_k)$
16: Update the model estimators and transmit it to the CSP: $\mathsf{ct}_{\beta^{t+1}} \leftarrow \beta^t - \mathsf{ct}_{\mathsf{GD}}$

[At the CSP]:
17: Decrypt the ciphertext $\mathsf{ct}_{\beta^{t+1}}$ and send the update model estimator β^{t+1} back to the server

[At the cloud server]:
18: Disseminate the updated estimator to each local site
19: **end for**
20: **return** β^t (last converged estimate)

and sends its encryption to the server (Step 12). For simplicity, we will omit the subscript t when it is clear from the context. The server aggregates g_k over encryption, yielding a ciphertext that encrypts a plaintext approximating to the global updated gradient $g = \sum_k g_k$ (Step 14). Next, it performs a homomorphic multiplication between the resulting ciphertext and the pre-computed inverse Hessian ciphertext, and outputs a ciphertext $\mathsf{ct}_{\mathsf{GD}}$ that represents a plaintext approximating to $\mathsf{Dec}(\mathsf{Inv}(\sum_k \mathsf{Enc}(\tilde{H}_k)) \cdot \sum_k \mathsf{Enc}(g_k)) \approx \tilde{H}^{-1} \cdot g$. The server securely updates the model estimator using the model parameters β^t and the encrypted gradient direction $\mathsf{ct}_{\mathsf{GD}}$, yielding the encrypted updated model estimator $\mathsf{ct}_{\beta^{t+1}}$ (Step 16). The server sends the resulting ciphertext to the CSP. After it is decrypted with the secret key of the CSP, it is sent back to the server while ensuring its privacy via DP. Later, we present a detailed proof that the estimated parameters are differentially private in Section IV-C. Finally, the server disseminates the perturbed model estimator to all the local sites. Here, the decryption procedure in the CSP has an important benefit in that the ciphertext modulus does not stack up, which significantly impacts computational efficiency. This is because the next update of the local gradient can be done using the plaintext vector β in the clear and each site can generate its encryption as a fresh input for the next iteration (Step 12). We describe this idea in more detail below in Section V-B.

B. Updated Diagonal Hessian Method

We introduce the diagonal updating approach via quasi-Cauchy relation to approximate diagonal Hessian [50]. This was originally motivated by the least change secant updating approach, in which the diagonal approximation is to be the sum of two diagonal matrices where the first diagonal matrix carries information of the local Hessian, while the second diagonal matrix is chosen so as to induce positive definiteness of the diagonal approximation. Because this approach updates the Hessian matrix at each iteration, it is more robust to the choice of the parameter λ.

We denote by $\mathsf{tr}(\cdot)$ the trace operator and let Ψ be a positive definite diagonal matrix. The update formulation for

Algorithm 2 U-SPLR: Homomorphic Evaluation of the Updated Diagonal Hessian Method

Input: Initial $\beta^0 = 0$; regularization parameter λ; DP budget ϵ; number of iterations of regression ι
Output: Global estimated coefficients β

[At the CSP]:
Generate keys: $(\mathsf{sk}, \mathsf{pk}, \mathsf{evk})$
[At local sites]:
for each site $k = 1$ to K **do**
  Generate a local random noise b_k
  Encrypt the local sample size n_k and transmit $\mathsf{Enc}(n_k)$ to the server
end for
[At the cloud server]:
Aggregate local sample sizes across sites and transmit to the CSP: $\sum_k \mathsf{Enc}(n_k)$
[At the CSP]:
Decrypt the ciphertext and disseminate the global sample size to each local site: $n \leftarrow \mathsf{Dec}(\sum_k \mathsf{Enc}(n_k))$
for $t = 0$ to $\iota - 1$ **do**
  [At local sites]:
  for each site $k = 1$ to K **do**
    Compute the local updated gradient g_k, the local vector V_k, and the local constant c_k
    Encrypt and transmit the ciphertexts to the server
  end for
  [At the cloud server]:
  Compute an encrypted constant for positive definiteness of diagonal Hessian: $\mathsf{ct}_\theta \leftarrow \textsc{CompTheta}(\mathsf{Enc}(c_k))$
  Aggregate the local vectors across sites: $\sum_k \mathsf{Enc}(V_k)$
  Compute the global vector W (Equation 16)
  Compute the global updated diagonal Hessian (Equation 11): $\mathsf{ct}_D \leftarrow \sum_k \mathsf{Enc}(V_k) - \mathsf{ct}_\theta \cdot W$
  Invert the global updated diagonal Hessian: $\mathsf{Inv}(\mathsf{ct}_D)$
  Aggregate local updated gradients across sites: $\sum_k \mathsf{Enc}(g_k)$
  Multiply the global gradient by the inverse Hessian: $\mathsf{ct}_{\mathsf{GD}} \leftarrow \mathsf{Inv}(\mathsf{ct}_D) \cdot \sum_k \mathsf{Enc}(g_k)$
  Update the model estimators and transmit it to the CSP: $\mathsf{ct}_{\beta^{t+1}} \leftarrow \beta^t - \mathsf{ct}_{\mathsf{GD}}$
  [At the CSP]:
  Decrypt the ciphertext $\mathsf{ct}_{\beta^{t+1}}$ and send the update model estimator β^{t+1} back to the server
  [At the cloud server]:
  Disseminate the updated estimator to each local site
end for
return β^ι (last converged estimate)

approximating Hessian matrix diagonally is derived as follows:

$$\begin{aligned} D_{t+1} &= \Psi + \left(\theta_t I + \frac{s_t^T u_t - s_t^T \Psi s_t - \theta_t s_t^T s_t}{\mathsf{tr}(E_t^2)} E_t\right) \\ &= \Psi + \frac{s_t^T u_t - s_t^T \Psi s_t}{\mathsf{tr}(E_t^2)} E_t - \theta_t \cdot \left(\frac{s_t^T s_t}{\mathsf{tr}(E_t^2)} E_t - I\right) \\ &= \textstyle\sum_{k=1}^K V_{tk} - \theta_t \cdot W_t, \end{aligned} \tag{11}$$

where

$$s_t = \beta^{t+1} - \beta^t, \tag{12}$$

$$\begin{aligned} u_t &= \textstyle\sum_k u_{tk} \\ &= \textstyle\sum_k (Z_k^T [y_k - \mu_k^{t+1}] - \frac{\lambda}{K}\beta^{t+1}) \\ &\quad - \textstyle\sum_k (Z_k^T [y_k - \mu_k^t] - \frac{\lambda}{K}\beta^t), \end{aligned} \tag{13}$$

$$E_t = \mathrm{diag}(s_{t,1}^2, \ldots, s_{t,d}^2), \tag{14}$$

$$V_{tk} = \frac{\Psi}{K} + \frac{s_t^T u_{tk}}{\mathsf{tr}(E_t^2)} E_t - \frac{s_t^T \Psi s_t}{K \cdot \mathsf{tr}(E_t^2)} E_t, \tag{15}$$

$$W_t = \frac{s_t^T s_t}{\mathsf{tr}(E_t^2)} E_t - I. \tag{16}$$

We denote by $s_{t,i}$ the i-th component of the vector s_t. See [50, Theorem 1] for more details. Since maintaining the positive definiteness for D_{t+1} is crucial for the quasi-Newton setting, the following lemma suggests a possible choice on θ_t.

Lemma 1 ([50, Lemma 2]): Assume that $s_t \neq 0$ for all t. Then D_t is positive definite, if

$$\theta_t = \min\{1, \frac{s_t^T u_t - s_t^T \Psi s_t}{s_t^T s_t}\}. \tag{17}$$

Note that the second input of the minimum function, denoted by c_t, can be computed in a distributed manner across local institutions as follows: $c_t = \sum_k c_{tk}$ where

$$c_{tk} = \frac{s_t^T u_{tk}}{s_t^T s_t} - \frac{s_t^T \Psi s_t}{K \cdot s_t^T s_t}. \tag{18}$$

Our secure and differentially private distributed logistic regression using the updated diagonal Hessian approximation is presented in Algorithm 2, denoted by U-SPLR. As mentioned before, this algorithm updates the global diagonal Hessian and computes its inverse at each iteration, so the

Algorithm 3 Homomorphic Computation for the Parameter θ

procedure COMPTHETA($\mathsf{Enc}(c_k)$)
[At the cloud server]:
1: Generate two random numbers ξ_1 and ξ_2, and encrypt the random number $(1+\xi_1)\cdot\xi_2$
2: Aggregate local constants across sites, add ξ_1, and multiply it by ξ_2: $(\sum_k \mathsf{Enc}(c_k)+\xi_1)\cdot\xi_2$
3: Transmit two ciphertexts to the CSP
[At the CSP]:
4: Decrypt two ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2$ and find the minimum value between them: $\mathsf{M} = \min\{\mathsf{Dec}(\mathsf{ct}_1), \mathsf{Dec}(\mathsf{ct}_2)\}$
5: Re-encrypt the value M and transmit it to the server
[At the cloud server]:
6: Multiply the ciphertext by ξ_2^{-1} and subtract ξ_1
7: **return** ciphertext ct_θ

pre-processing phase of U-SPLR is done in a simpler way than F-SPLR.

At each iteration, each local institution updates the local gradient g_k using its local noise b_k and the sample size n as in (10). Each site also computes the local vector V_k and the local statistics c_k as in (15) and (18), respectively. After that, it encrypts all the values and sends the output ciphertexts $\mathsf{Enc}(g_k)$, $\mathsf{Enc}(V_k)$, and $\mathsf{Enc}(c_k)$ to the server (Step 10 and 11 in Algorithm 2).

The server begins by computing an encryption of the constant $\theta = \min\{1, \sum_k c_k\}$, which can be used for ensuring positive definiteness of the updated diagonal Hessian (Step 13 in Algorithm 2 or COMPTHETA($\cdot$) in Algorithm 3). We employ a random masking technique to obscure the inputs while computing their minimum. At first, the server generates two positive random numbers $\xi_1, \xi_2 < 1$ and obtains two ciphertexts: one is the encryption of $(1+\xi_1)\cdot\xi_2$ and the other is computed by $(\sum_k \mathsf{Enc}(c_k)+\xi_1)\cdot\xi_2$. The output ciphertexts are sent to the CSP, say ct_1 and ct_2 (Step 1 to 3 in Algorithm 3). It follows from the IND-CPA security of the underlying HE scheme that the ciphertexts are computationally indistinguishable. The CSP learns nothing else after decryption while computing the (scaled) minimum value between them (Step 4), say $\mathsf{M} = \min\{\mathsf{Dec}(\mathsf{ct}_1), \mathsf{Dec}(\mathsf{ct}_2)\}$. The CSP re-encrypts M and sends it to the server (Step 5). Using the fact that

$$\begin{aligned}\mathsf{M} &= \min\{\mathsf{Dec}(\mathsf{Enc}((1+\xi_1)\cdot\xi_2)),\\ &\qquad \mathsf{Dec}((\textstyle\sum_k \mathsf{Enc}(c_k)+\xi_1)\cdot\xi_2)\}\\ &\approx \min\{(1+\xi_1)\xi_2, (\textstyle\sum_k c_k+\xi_1)\xi_2\}\\ &= \big(\min\{1, \textstyle\sum_k c_k\}+\xi_1\big)\cdot\xi_2, \end{aligned} \tag{19}$$

the value of θ can be recovered from M as follows:

$$\theta = \min\{1, \textstyle\sum_k c_k\} \approx \xi_2^{-1}\cdot\mathsf{M} - \xi_1. \tag{20}$$

In other words, the server performs homomorphic evaluation of (20) to get a ciphertext of the desired constant θ:

$$\mathsf{ct}_\theta = \xi_2^{-1}\cdot\mathsf{Enc}(\mathsf{M}) - \xi_1. \tag{21}$$

Now the server securely aggregates V_k over encryption and computes the global vector W using (16) (Step 14 and 15 in Algorithm 2). Equation (11) can be homomorphically evaluated to obtain the global updated diagonal Hessian ciphertext ct_D, that is, $\mathsf{ct}_D = \sum_k \mathsf{Enc}(V_k) - \mathsf{ct}_\theta \cdot W$. Note that

$$\begin{aligned}\mathsf{Dec}(\mathsf{ct}_D) &\approx \mathsf{Dec}(\textstyle\sum_k \mathsf{Enc}(V_k)) - \mathsf{Dec}(\mathsf{ct}_\theta \cdot W)\\ &\approx \textstyle\sum_k V_k - \theta\cdot W = D. \end{aligned} \tag{22}$$

Then the server carries out the inversion operation of the output ciphertext ct_D (Step 16 and 17). Next, it aggregates g_k over encryption and performs a homomorphic multiplication between the resulting ciphertext and the updated inverse Hessian ciphertext ct_D (Step 19). This procedure outputs a ciphertext $\mathsf{ct}_{\mathsf{GD}}$ that encrypts a plaintext approximating to $\mathsf{Dec}(\mathsf{Inv}(\mathsf{ct}_D)\cdot\sum_k \mathsf{Enc}(g_k)) \approx D^{-1}\cdot g$. Similar to F-SPLR, the server securely updates the model estimator using β^t and $\mathsf{ct}_{\mathsf{GD}}$, yielding the encrypted updated model estimator $\mathsf{ct}_{\beta^{t+1}}$ (Step 20). The server sends the resulting ciphertext to the CSP. After it is decrypted by the CSP, the updated model estimator is sent back to the server and disseminated to all the sites.

C. Correctness

Theorem 1: Given a set of n examples $x_1, \ldots, x_n$ over $\mathbb{R}^d$ with labels $y_1, \ldots, y_n$ such that $||x_i|| \le 1$ for each i, the output model estimator of each iteration in Algorithm 1 or Algorithm 2 preserves (ϵ/ι)-DP.

Proof: Following the proof of Theorem 2 in [11], we assume α and α' are two normalized vectors over $\mathbb{R}^d$, and $y, y' \in \{-1, 1\}$ are the corresponding binary labels. For every unique b (associated with the added perturbation term $b^T\beta/n$ to the objective), there is an one-to-one mapping between the inputs S and output β^* because the regularization function and loss function are differentiable everywhere. Consider $S_\alpha = \{(z_1, y_1), \ldots, (z_{n-1}, y_{n-1}), (\alpha, y)\}$ and $S_{\alpha'} = \{(z_1, y_1), \ldots, (z_{n-1}, y_{n-1}), (\alpha', y')\}$, which differ only by the last instance. We can sample b_1 and b_2 from the Laplacian distribution so that these two different log-likelihood functions share the same β^*. That is, for any b_1, there exists b_2 such that $b_1 - \frac{y\alpha}{1+e^{y(\beta^*)^T\alpha}} = b_2 - \frac{y'\alpha'}{1+e^{y'(\beta^*)^T\alpha'}}$. Because $||\alpha||, ||\alpha'|| \le 1$, and $\frac{1}{1+e^{y'(\beta^*)^T\alpha'}}, \frac{1}{1+e^{y(\beta^*)^T\alpha}} \le 1$, we get $||b_1 - b_2|| \le 2$ for any β^*, and therefore, $||b_1|| - 2 \le ||b_2|| \le ||b_1|| + 2$.

The outputs of the Algorithm 1 and Algorithm 2 are the updated β^+ following the gradient direction before converging to β^*. We show in the Appendix A that the difference of two log-likelihood functions (induced by (S_α, b_1) and (S'_α, b_2)) has a single optimum at β^*. Using this property together with $||b_1|| - 2 \le ||b_2|| \le ||b_1|| + 2$, we can show that

$$\begin{aligned}&\frac{\Pr[\beta^+|z_1, \ldots, z_{n-1}, y_1, \ldots, y_{n-1}, z_n = \alpha, y_n = y]}{\Pr[\beta^+|z_1, \ldots, z_{n-1}, y_1, \ldots, y_{n-1}, z_n = \alpha', y_n = y']}\\ &\qquad \le e^{\epsilon/2\iota(||b_1|| - ||b_2||)} \end{aligned} \tag{23}$$

is at most $e^{\epsilon/\iota}$, as desired. □

We refer the reader to the Appendix A for details.

V. EXPERIMENTS

We conducted experiments to validate our proposed methods on two real datasets. For comparison, we set the prediction performance of the global model with the entire samples as

TABLE I
DESCRIPTION OF DATASETS

Dataset	n	K	d	IR
PhysioNet Challenge 2012	4,000	5	189	6.22
Diabetes 130-US hospitals	69,974	10	40	1.46

a reference. Let us assume that there are K sites with the same number of patients n_k for simplicity, but each site has no information about other local sample sizes. We utilized κ-fold cross validation (CV) that randomly splits patients into κ folds with the equal size; we used $(\kappa-1)$ folds for training and one fold for testing. As an evaluation measure, we used the area under the receiver operating characteristic curve (AUC), which is a summarized single value for the curve. AUC has desirable properties that are independent to a threshold and invariant to a priori class probability distributions. Therefore, it is a widely used measure to evaluate accuracy in classification problems of imbalanced data and both datasets that we used are imbalanced. All the experiments were performed on a machine with an Intel Core i7 running at 2.9 GHz processor. We used the approximate HE scheme of [48]. As mentioned in Section III-C, we take advantage of the ciphertexts packing method and the homomorphic SIMD technique to perform computation in parallel.

A. Data Description

We used two datasets: PhysioNet Challenge 2012 [51] and Diabetes 130-US hospitals [52]. PhysioNet Challenge 2012 dataset extracted from the Multiparameter Intelligent Monitoring in Intensive Care II database comprised of patient stays in the intensive care unit (ICU) lasting at least 48 hours to predict mortality. We used the dataset A consisting of 4000 subjects whose age at ICU admission was 16 years or over. The data were formatted as time-stamped measurements for 37 distinct variables and four static variables. We transformed each time-series variable into min, max, mean, first value, and last value variables as a way to summarize it. Missing values were replaced by the mean value of a variable. On the other hand, Diabetes 130-US hospitals dataset was developed to study factors related to readmission for patients with diabetes. The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks with over 50 features representing patient and hospital outcomes. We preprocessed data following the work of Strack *et al.* [52]. Table I illustrates the datasets with the number of observations (n), the number of local institutions (K), and the number of features (d), the imbalance ratio (IR), respectively. IR is defined as the ratio of the number of instances in the majority class to the number of examples in the minority class. Note that we used 5-fold CV for the first dataset and 10-fold CV for the other.

B. HE-parameters Tuning

Cheon *et al.* [48] introduced an approximation to convert the multiplicative inverse function into a low-degree polynomial and presented an efficient evaluation strategy. The only constraint ensuring the convergence of the geometric series in the approximation is that an input of the evaluation function should be in the range of $[1/2, 3/2]$. We observe that

$$(\sum \tilde{H}_k)^{-1} \cdot (\sum g_k) = (\sum \tfrac{\tilde{H}_k}{\gamma})^{-1} \cdot (\sum \tfrac{g_k}{\gamma}) \tag{24}$$

for a scaling factor γ. In our implementation, the local diagonal Hessian $\tilde{H}_k$ is divided by γ before encryption so that $\tilde{H}_k/\gamma$ is in the range and one can guarantee its convergence.

Suppose that we adapt the degree (2^r-1) approximating polynomial of the inverse function. Specifically, the arithmetic circuit can be expressed as $\mathsf{Inv}(x) = \prod_{j=0}^{r-1}(1+\bar{x}^{2^j})$ where $\bar{x} = 1-x$. It is a common practice to perform the rescaling procedure by a factor of p on ciphertexts after each multiplication in order to maintain the precision of plaintext. Thus, the ciphertext modulus is reduced by $\log p$ bits after a multiplication. As discussed in [48], the evaluation of the above equation requires a total ciphertext modulus of $r \log p$ bits. In the case of F-SPLR, it requires $\log p$ more bits to perform a multiplication between the encrypted inverse Hessian and the encrypted global gradient. In the case of U-SPLR, it requires $2\log p$ more bits than F-SPLR: $\log p$ bits for a multiplication by the constant ξ_2^{-1} (Step 6 in Algorithm 3) and $\log p$ bits for a multiplication between the ciphertext ct_θ and the global vector W (Step 16 in Algorithm 2). Therefore, a lower-bound on the bit length of a fresh ciphertext modulus, denoted by $\log q$, is as follows:

$$\begin{cases} (r+1)\log p + \log q_0, & \text{for F-SPLR} \quad (25a) \\ (r+3)\log p + \log q_0, & \text{for U-SPLR} \quad (25b) \end{cases}$$

where q_0 is the output ciphertext modulus. The final ciphertext represents the desired vector β but is scaled by a factor of p, which implies that $\log q_0$ should be larger than $\log p$.

We note that the HEAAN scheme comes with encoding/encryption errors as well as computation error. The first type errors are $O(N)$ and the precision loss during computation is bounded by a depth of a circuit (see the detail in [48, Lemma1]). For example, the precision loss during the evaluation of F-SPLR exceeds at most $(r+1)$ bits when compared to the unencrypted approximate computation. So, we multiply the scale factor p to input messages before encryption to reduce the precision loss from the resulting errors. In our implementation, we assume that all the inputs have $\log p = 35$ bits of precision and set the bit length of the output ciphertext modulus as $\log q_0 = \log p + 20$.

We further optimized the U-SPLR algorithm by manipulating the arithmetic circuit for computing the ciphertext ct_D. Specifically, it can be expressed as follows:

$$\begin{aligned} \mathsf{ct}_\theta \cdot W &= (\xi_2^{-1} \cdot \mathsf{Enc}(\mathsf{M}) - \xi_1) \cdot W \\ &= (\xi_2^{-1} \cdot W) \cdot \mathsf{Enc}(\mathsf{M}) - \xi_1 \cdot W. \end{aligned} \tag{26}$$

If the server multiplies ξ_2^{-1} and W in plaintext, it only needs to multiply the ciphertext $\mathsf{Enc}(\mathsf{M})$ by the resulting constant $(\xi_2^{-1} \cdot W)$. As a result, this method could reduce the bit length of a fresh ciphertext modulus down to $(r+2)\log p + \log q_0$.

In our implementation, we used the Gaussian distribution of standard deviation $\sigma = 3.2$ to sample error polynomials and we set $h = 64$ as the number of nonzero coefficients

TABLE II
HE PARAMETER SET

Method	N	$\log p$	$\log q$	Ciphertext size
F-SPLR	2^{14}	35	230	0.89 MB
U-SPLR			265	1.03 MB

TABLE III
AVERAGED GLOBAL AUCS OF F-LR AND U-LR WITH STANDARD DEVIATIONS (PARENTHESIS)

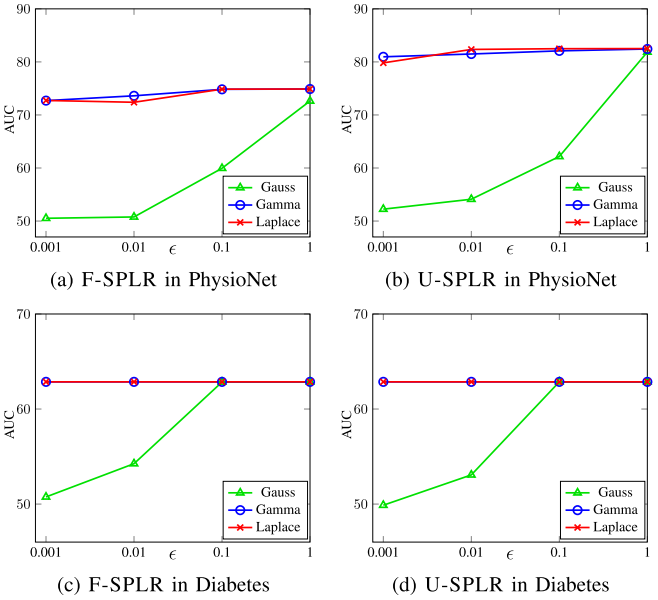
	F-LR			U-LR		
λ	10	100	500	10	100	500
T	300	100	50	50		
PhysioNet	0.749 (0.020)	0.711 (0.019)	0.727 (0.019)	0.828 (0.021)	0.830 (0.024)	0.802 (0.041)
Diabetes	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)

in a secret key $s(X)$. We took the ring dimension $N = 2^{14}$ to ensure 80 bits of security against the known attacks on the LWE problem and we checked its security by using the LWE estimator of Albrecht *et al.* [53]. We set $r = 4$, that is, we used the degree 15 approximating polynomial to the inverse function and we could actually obtain the bit length of a fresh ciphertext modulus as 230 and 265 for F-SPLR and U-SPLR, respectively. For these settings, the key generation takes about two seconds and the encryption takes around one second. In Table II, we summarize the parameter setting as well as the sizes of the public key and a freshly encrypted ciphertext.

C. Experimental Results

1) Accuracy Analysis: In the rest of the paper, we let T denote the total number of iterations for the Newton-Raphson method. Table III shows results of the global logistic regression models with fixed and updated diagonal Hessian approximations without HE and DP, denoted by F-LR and U-LR, respectively. These results are the references for the comparison with DLPA. Because F-SPLR is relatively sensitive to λ, it is necessary to differentiate T depending on λ to get a good performance. From the preliminary experiments, we determined the iteration number T for each λ as shown in Table III and this setting was equally applied to other experiments.

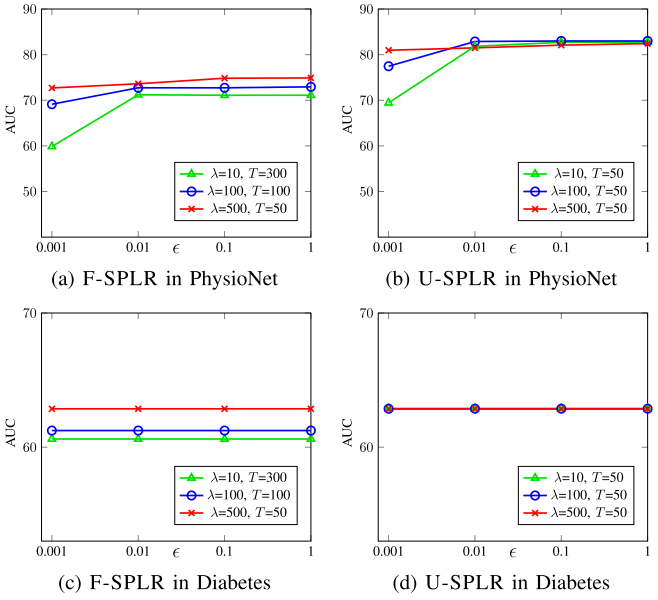
We see that F-SPLR has comparable results with those pairs; the larger the lambda, the smaller numbers of iterations are needed in both datasets. For U-SPLR, we set T to the smallest one in U-SPLR without any change. We note that we set the DP budget ϵ to ϵ/T because the budget at every iteration is accumulated into the total budget. Despite setting the parameters in favor of F-SPLR, U-SPLR outperforms F-SPLR in terms of AUC. The gap between two methods is more significant in PhysioNet dataset than in the Diabetes one. Fig. 2 shows the effect of Gauss, Gamma, and Laplace DLPA on the accuracy of F-SPLR and U-SPLR with fixed values of λ and T. It is encouraged that F-SPLR and U-SPLR based on Gauss, Gamma, and Laplace DLPA reach their global



(a) F-SPLR in PhysioNet (b) U-SPLR in PhysioNet

(c) F-SPLR in Diabetes (d) U-SPLR in Diabetes

Fig. 2. (a), (b) Averaged AUCs of F-SPLR and U-SPLR based on Gauss, Gamma, and Laplace DLPA with λ = 500, T = 50 in PhysioNet; (c), (d) Averaged AUCs of F-SPLR and U-SPLR based on Gauss, Gamma, and Laplace DLPA with λ = 500, T = 50 in Diabetes.



(a) F-SPLR in PhysioNet (b) U-SPLR in PhysioNet

(c) F-SPLR in Diabetes (d) U-SPLR in Diabetes

Fig. 3. (a), (b) Averaged AUCs of F-SPLR and U-SPLR based on Gamma DLPA with different combinations of λ and T in PhysioNet; (c), (d) Averaged AUCs of F-SPLR and U-SPLR based on Gamma DLPA with different combinations of λ and T in Diabetes.

accuracies as ϵ increases. However, Gamma and Laplace DLPA are stable enough even at small ϵ, while Gauss DLPA is not.

Fig. 3 shows the effect of different combinations of parameters λ and T on the accuracy of F-SPLR and U-SPLR based on Gamma DLPA. What we can see in this experiment is that U-SPLR is robust to λ but F-SPLR is sensitive to λ. Therefore, to obtain good results in F-SPLR, T should be adjusted according to λ as in the global model. In particular,

TABLE IV
COMPLEXITIES OF F-SPLR AND U-SPLR FOR $T = 50$. NUMBERS ARE IN SECONDS AND STANDARD DEVIATIONS ARE IN THE PARENTHESES

		F-SPLR			U-SPLR		
		Local	Server	CSP	Local	Server	CSP
Theoretical complexity	Preparation	$2K\mathsf{Enc}$	$2r\mathsf{HM}$	$1\mathsf{Dec}$	$K\mathsf{Enc}$	–	$1\mathsf{Dec}$
	Iteration	$K\mathsf{Enc}$	$1\mathsf{HM}$	$1\mathsf{Dec}$	$3K\mathsf{Enc}$	$1\mathsf{Enc} + 2r\mathsf{HM}$	$3\mathsf{Dec} + 1\mathsf{Enc}$
	Total	$\boldsymbol{KT\mathsf{Enc}}$	$\boldsymbol{(T+2r)\mathsf{HM}}$	$\boldsymbol{T\mathsf{Dec}}$	$\boldsymbol{3KT\mathsf{Enc}}$	$\boldsymbol{T\mathsf{Enc} + (2rT)\mathsf{HM}}$	$\boldsymbol{3T\mathsf{Dec} + T\mathsf{Enc}}$
Experimental complexity of PhysioNet	Preparation	0.685s (0.006)	0.715s (0.013)	<0.001s (<0.001)	0.360s (0.005)	–	<0.001s (<0.001)
	Iteration	0.343s (0.003)	0.115s (0.001)	<0.001s (<0.001)	1.079s (0.015)	1.042s (0.005)	0.171s (0.004)
	Total	**17.947s** (0.274)	**6.502s** (0.109)	**0.005s** (<0.001)	**54.307s** (0.769)	**52.121s** (0.261)	**8.551s** (0.204)
Experimental complexity of Diabetes	Preparation	1.178s (0.008)	0.713s (0.005)	<0.001s (<0.001)	0.638s (0.010)	–	<0.001s (<0.001)
	Iteration	0.590s (0.003)	0.121s (0.002)	<0.001s (<0.001)	1.913s (0.029)	1.038s (0.009)	0.137s (0.003)
	Total	**30.691s** (0.412)	**6.812s** (0.150)	**0.005s** (<0.001)	**96.271s** (1.444)	**51.892s** (0.428)	**9.153s** (0.197)

F-SPLR has the best accuracy at $\lambda = 500$ and $T = 50$ in both datasets, but it can never outperform U-SPLR. It should be noted that only the result of Gamma DLPA is provided here for simplicity, but the trend in the result remains the same in Gauss and Laplace DLPA. We describe more detailed results in Appendix B. In addition, it is found that parameter estimates (i.e., coefficients β) have the same trend of AUC. That is, when epsilon increases, the coefficients obtained from experiments in a secure and privacy-preserving manner reach within a reasonable range of estimates of the global model.

2) Complexity Analysis: Homomorphic multiplication is the most time-consuming procedure, so in the following description we will only count the number of the operations. We note that homomorphic evaluation of the degree $(2^r - 1)$ approximating polynomial of the inverse function requires $2(r-1)$ homomorphic multiplications (see Section V-B).

In the case of F-SPLR, the local site encrypts its own local gradients at each iteration (Step 12 in Algorithm 1). The server securely aggregates K number of the local gradients, multiplies it by the pre-computed inverse Hessian, and updates the model estimator. The resulting ciphertext $\mathsf{ct}_{\beta^{t+1}}$ is decrypted by the CSP. In total, K local institutions encrypt the information $K \cdot (T+2)$ times, the server takes $T + 2(r-1)$ multiplications, and the CSP performs $(T+1)$ decryptions. In Table IV, we summarize the theoretical complexity where HM denotes homomorphic multiplication. In addition, we report experimental results with $T = 50$. In Appendix C, we provide more results at various numbers of iterations for F-SPLR.

In the case of U-SPLR, the local site encrypts three types of data at each iteration: the local constant c_k, the local vectors V_k, and the local updated gradients g_k, which means that each site takes about three times more effort to encrypt the data. In order to securely compute the constant θ, the server generates an encryption of a random value, and the CSP decrypts two ciphertexts and re-encrypts the minimum result. After that, the server takes $2(r-1)$ homomorphic multiplications to compute the inverse of the global diagonal Hessian (Step 17 in Algorithm 2) and needs one more multiplication to compute the increment (Step 19). Finally, the CSP needs to decrypt the resulting ciphertext $\mathsf{ct}_{\beta^{t+1}}$ (Step 21). In total, K local institutions encrypt the intermediate statistics $K \cdot (3T+1)$ times, the server performs one encryption operations and takes $(2r-1) \cdot T$ multiplications, and the CSP performs $(3T+1)$ decryptions and T encryptions.

TABLE V
CIPHERTEXT SIZES OF F-SPLR AND U-SPLR

	F-SPLR	U-SPLR
Each local site $\rightarrow$ server	46.28 MB	155.53 MB
Server $\rightarrow$ CSP	11.39 MB	115.05 MB
CSP $\rightarrow$ server	–	51.76 MB

As shown in Table IV, the running timings of the server and CSP are similar in both datasets. However, Diabetes dataset has more local institutions than PhysioNet, so, it takes $K_{\text{Diabetes}}/K_{\text{PhysioNet}} = 2$ times more to encrypt the Diabetes dataset than PhysioNet. We note that the local encryptions can be done in a synchronous way.

3) Communication Complexity: At each iteration of U-SPLR, each local site generates two more ciphertexts ($\mathsf{Enc}(c_k)$, $\mathsf{Enc}(V_k)$) compared to F-SPLR, and sends them to the server. So it results in a larger communication cost between the local institutions and the server. In U-SPLR, the server sends two randomized ciphertexts to get the ciphertext ct_θ to the CSP and CSP transmits the encrypted minimum value $\mathsf{Enc}(\mathsf{M})$ to the server. Table V summarizes the numbers between the institutions involved in our protocols.

VI. DISCUSSION

A. Hybrid Model of Distributed Logistic Regression

HE provides a practical solution for secure data outsourcing, but it can only evaluate functions of fixed depth. This is because a ciphertext modulus decreases or a level of noise grows with operations, and it finally becomes too small or large to carry out more computation. Previous studies [30], [31] are based on the *leveled* HE scheme for efficiency, so they have an inherent challenge such that it only allows a very small number of iterations of gradient descent training. Because it converges very slowly, their methods are not amenable to get accurate models for all datasets. Recently, Cheon *et al.* [54] proposed a new technique to

refresh low-level ciphertexts in the HEAAN scheme and keep computing on encrypted data. However, this procedure still entails expensive computational cost, so it is a non-trivial task to apply them to real-world applications.

This article proposes a new method to use a fruitful combination of HE and DP for an iterative learning algorithm. If we perturb data with noise and it does not significantly change the estimation, one can decrypt it and use the information for regression while ensuring the privacy. In our implementation, we decrypted the model estimator at each iteration and it was re-distributed to generate fresh inputs for the next iteration. It enables us to use smaller HE parameters than the ones used in previous approaches [30], [31], and consequently, this leads to better performance in terms of computation efficiency and accuracy.

A similar approach was introduced by Aono *et al.* [32], which aims to securely compute a polynomial approximation of the objective function and directly add noise to coefficients of the approximation. As a result, the secret-key owner can get the perturbed coefficients of the objective function after decryption. Their approach is computationally efficient, but it tends to incur accuracy loss because the scale of noise is roughly proportional to d^r when taking a degree-r polynomial approximation with d-dimensional input ([32, Theorem 2]). In comparison, our method would be more useful for learning tasks with many features because its accuracy is not affected by the number of features.

B. Comparison With F-SPLR and U-SPLR

In this paper, we presented two types of diagonal Hessian approximation methods: F-SPLR and U-SPLR. As mentioned in Section IV-A, the latter algorithm repeatedly updates the approximate inverse Hessian via the quasi-Cauchy relation, so at each iteration, it has one more interaction with the CSP to compute the value for maintaining positive definiteness of Hessian and needs more homomorphic computation for the update. When $r = 4$, the computational costs of the server for F-SPLR and U-SPLR are $(T+6)$ and $7T$ homomorphic multiplications, respectively. That is, the server performs around 7 times more computations to proceed U-SPLR than F-SPLR. However, U-SPLR is more robust to the user-defined parameter λ than F-SPLR, and thus U-SPLR shows stable results with the constant number of iterations regardless of λ rather than F-SPLR. The difference between AUCs of U-SPLR and F-SPLR is also more significant when the data dimensionality is high. As a result, we recommend F-SPLR if there is prior knowledge of λ, otherwise U-SPLR.

C. Comparison with Gauss, Gamma, and Laplace DLPA

Our experimental results show that AUCs of all three mechanisms approach the global AUC as the privacy budget ϵ increases regardless of the diagonal approximation methods. However, the AUC from the Gauss mechanism is significantly lower than the others when ϵ is relatively small. This is because the Gauss mechanism generates significantly more redundant noise, which brings degradation of its accuracy. The Gamma and Laplace mechanisms generate similar amounts of redundant noise for any ϵ, which results in similar patterns in accuracy. These findings are consistent with the work of Goryczka and Xiong [44].

D. Limitations and Future Work

We demonstrated the feasibility of secure and differentially private logistic regression and the experiments were conducted on a single machine (with different processes) to serve as a proof-of-concept. In practice, we need to deploy the algorithm in multiple computers.

U-SPLR has an advantage of robustness in terms of learning rate and λ, and it can converge in fewer iterations when compared to existing algorithms based on the gradient descent method. Nevertheless, it should be noted that the accuracy of our methods can be decreased as T and/or K increase(s) because the noise distribution is dependent on these two parameters (see Appendix D for details). In particular, because T is a fixed priori and ϵ is evenly split across iterations in our algorithms, accuracy highly depends on T. As T is increased, the privacy budget for each iteration becomes smaller, which adds a lot of noise to each gradient and thus results in accuracy degradation. To address this problem, we will consider the dynamic allocation of privacy budget for each iteration [55] to do intensive analysis on accuracy and T in future work. The optimal values of parameters, λ and T, are data-dependent, which makes hard to select those a priori. There is definitely a need to study this problem using more principled approaches.

Although there are many extensions of the objective perturbation approach, we used the straightforward objective perturbation technique to demonstrate the feasibility of integrating HE and DP in a harmonious manner. We will consider using these extensions to do more precise and rigorous analysis of DP at applicable scenarios.

VII. Conclusion

We develop a novel secure and differentially private distributed logistic regression model by harmonizing state-of-the-art security and privacy frameworks. To the best of our knowledge, this is the first work that combines DP and HE to support the construction of a distributed learning model. Combining the complementary strength of both frameworks, our solutions achieved high efficiency and good accuracy.

Appendix A
Proof of Theorem 1

We assume that α and α' are two normalized vectors over $\mathbb{R}^d$, and $y, y' \in \{-1, 1\}$ are the corresponding binary labels. Consider two sets of inputs: $S_\alpha = \{(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), (\alpha, y)\}$ and $S_{\alpha'} = \{(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), (\alpha', y')\}$, where $x_n = \alpha$ or α' and $y_n = y$ or y' in the context of S_α or $S_{\alpha'}$. The negative log-likelihood function has the following form

$$l(\beta, b, x_n, y_n) = \frac{\lambda}{2}\beta^T\beta + \frac{b^T\beta}{n} + \frac{1}{n}\sum_{i=1}^{n-1}\log(1 + e^{-y_i\beta^T x_i}) + \frac{1}{n}\log(1 + e^{-y_n\beta^T x_n}) \quad (27)$$

where b is a random sample from the density function $h(b) \propto e^{-\frac{\epsilon}{2}||b||}$.

For any output β^* by our algorithm, there is a unique value of b that maps the input to the output because the entire function is differentiable everywhere. We pick a common minimizer β^* for both sets of inputs: S_α and $S_{\alpha'}$. Note that there is b_1 that maps these inputs S_α to β^*, and similarly, there is a unique b_2 that maps these inputs $S_{\alpha'}$ to β^*. Since β^* optimizes both problems, the derivatives at this point are 0, which means there exist b_1, b_2 such that

$$b_1 - \frac{y\alpha}{1+e^{y(\beta^*)^T\alpha}} = b_2 - \frac{y'\alpha'}{1+e^{y'(\beta^*)^T\alpha'}}. \tag{28}$$

Because $||\alpha||, ||\alpha'|| \leq 1$ and $\frac{1}{1+e^{y(\beta^*)^T\alpha}}, \frac{1}{1+e^{y'(\beta^*)^T\alpha'}} \leq 1$, we get $||b_1 - b_2|| \leq 2$ for any β^*, and therefore, $||b_1|| - 2 \leq ||b_2|| \leq ||b_1|| + 2$.

Now we look at the intermediate estimated parameters $\beta^+ = \beta^* - c$ (i.e., obtained in gradient based optimization before convergence), where c is a vector of the same dimension as β^*. Therefore, we know that

$$\begin{aligned}
&\frac{\Pr[\beta^+|x_1,\ldots,x_{n-1},y_1,\ldots,y_{n-1},x_n=\alpha,y_n=y]}{\Pr[\beta^+|x_1,\ldots,x_{n-1},y_1,\ldots,y_{n-1},x_n=\alpha',y_n=y']} \\
&\quad\propto \frac{\mu(b_1|S_\alpha)\cdot\exp\left(l(\beta^+,b_1,\alpha,y)\right)}{\mu(b_2|S'_\alpha)\cdot\exp\left(l(\beta^+,b_2,\alpha',y')\right)} \\
&\quad\propto \frac{\mu(b_1|S_\alpha)\cdot e^{\frac{\lambda}{2}(\beta^+)^T\beta^+}\cdot e^{\frac{b_1^T\beta^+}{n}}}{\mu(b_2|S'_\alpha)\cdot e^{\frac{\lambda}{2}(\beta^+)^T\beta^+}\cdot e^{\frac{b_2^T\beta^+}{n}}} \\
&\qquad\times\frac{\prod_{i=1}^{n-1}(1+e^{-y_i(\beta^+)^Tx_i})^{\frac{1}{n}}\cdot(1+e^{-y(\beta^+)^T\alpha})^{\frac{1}{n}}}{\prod_{i=1}^{n-1}(1+e^{-y_i(\beta^+)^Tx_i})^{\frac{1}{n}}\cdot(1+e^{-y'(\beta^+)^T\alpha'})^{\frac{1}{n}}} \\
&\quad\propto \frac{\mu(b_1|S_\alpha)\cdot e^{\frac{b_1^T\beta^+}{n}}\cdot(1+e^{-y(\beta^+)^T\alpha})^{\frac{1}{n}}}{\mu(b_2|S'_\alpha)\cdot e^{\frac{b_2^T\beta^+}{n}}\cdot(1+e^{-y'(\beta^+)^T\alpha'})^{\frac{1}{n}}}
\end{aligned} \tag{29}$$

where $\mu(b|D)$ is the densities of b given the output β^*. Note that Theorem 9 of [12] implies,

$$\begin{aligned}
\frac{\mu(b_1|S_\alpha)}{\mu(b_2|S'_\alpha)} &= \frac{||b_1||^{d-1}e^{-\epsilon||b_1||/2\iota}\cdot\frac{1}{surf(||b_1||)}}{||b_2||^{d-1}e^{-\epsilon||b_2||/2\iota}\cdot\frac{1}{surf(||b_2||)}} \\
&\leq e^{\epsilon(||b_1||-||b_2||)/2\iota} \leq e^{\epsilon/\iota}
\end{aligned} \tag{30}$$

where ι is the number of iterations in the optimization and $surf(x)$ denotes the surface area of the sphere in dimension d with radius x. Since two log-likelihood functions share the same minimizer β^*, we can make them meet by adding a scalar c, which does not change the shape of the log-likelihood functions and therefore preserves parameter estimation results. This adjustment makes the ratio of the likelihood functions equal 1, that is,

$$e^{\frac{b_1^T\beta^*}{n}}\cdot(1+e^{-y(\beta^*)^T\alpha})^{\frac{1}{n}} = e^{\frac{b_2^T\beta^*}{n}}\cdot(1+e^{-y'(\beta^*)^T\alpha'})^{\frac{1}{n}}.$$

When a point is deviating from the optimum (e.g. moving on the gradient ascending direction along any dimension), the likelihood ratio will change. Fig. 4 shows when each dimension of the estimated parameters is departing from the optimum β^* (i.e., replacing β^* with $\beta^+ = \beta^* - c$), the following inequality satisfies from the fact that the difference of the negative log-likelihood functions is greater or equals to 0. It can be shown that the derivative of the following function of any dimension of β

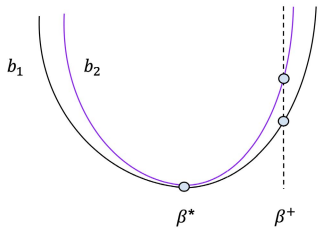


Fig. 4. Illustration on the inequality of the log-likelihood function at β^+.

TABLE VI
AVERAGED AUCs WITH STANDARD DEVIATIONS (PARENTHESIS) OF F-SPLR AND U-SPLR WITH GAUSS DLPA ON PhysioNet

	F-SPLR			U-SPLR		
	λ			λ		
	10	100	500	10	100	500
ϵ \ T	300	100	50	50		
0.001	0.508 (0.039)	0.511 (0.039)	0.505 (0.050)	0.487 (0.052)	0.519 (0.029)	0.522 (0.047)
0.01	0.512 (0.048)	0.513 (0.047)	0.508 (0.043)	0.501 (0.030)	0.523 (0.040)	0.541 (0.054)
0.1	0.514 (0.021)	0.526 (0.083)	0.600 (0.086)	0.550 (0.028)	0.575 (0.058)	0.622 (0.031)
1	0.622 (0.022)	0.709 (0.014)	0.726 (0.020)	0.774 (0.017)	0.829 (0.023)	0.819 (0.026)

$$\begin{aligned}
&n\cdot l(\beta,b_2,\alpha',y') - n\cdot l(\beta,b_1,\alpha,y) \\
&\quad= \log(1+e^{-y'\beta^T\alpha'}) - \log(1+e^{-y\beta^T\alpha}) \\
&\qquad+\frac{y'\beta^T\alpha'}{1+e^{y'(\beta^*)^T\alpha'}} - \frac{y\beta^T\alpha}{1+e^{y(\beta^*)^T\alpha}}
\end{aligned} \tag{31}$$

only equals to 0 when $\beta = 0$, which means there is a single minimum of the differences between the likelihood functions. When we plug $\beta = 0$ into the above equation, it equals 0 and the differences between the two functions are greater or equals to 0 at any point. Therefore, the following inequality holds on any dimension of β^+

$$e^{\frac{b_1^T\beta^+}{n}}\cdot(1+e^{-y(\beta^+)^T\alpha})^{\frac{1}{n}} \leq e^{\frac{b_2^T\beta^+}{n}}\cdot(1+e^{-y'(\beta^+)^T\alpha'})^{\frac{1}{n}}.$$

Finally, we have the following inequality

$$\frac{\Pr[\beta^+|x_1,\ldots,x_{n-1},y_1,\ldots,y_{n-1},x_n=\alpha,y_n=y]}{\Pr[\beta^+|x_1,\ldots,x_{n-1},y_1,\ldots,y_{n-1},x_n=\alpha',y_n=y']} \leq e^{\epsilon/\iota}.$$

This result is in line with that of Theorem 1 of [56], which shows the update of β can be protected by adding a Laplacian noise $\mu(b|D) \propto \frac{1}{n}e^{-\epsilon/2}$ to the gradient (if all the budgets are spent in a single iteration, i.e., $\iota = 1$).

APPENDIX B
PREDICTION PERFORMANCE

See Table VI to XI

TABLE VII

AVERAGED AUCs WITH STANDARD DEVIATIONS (PARENTHESIS) OF F-SPLR AND U-SPLR WITH GAMMA DLPA ON PhysioNet

	F-SPLR			U-SPLR		
	λ			λ		
	10	100	500	10	100	500
ϵ \ T	300	100	50	50		
0.001	0.599 (0.031)	0.691 (0.010)	0.727 (0.017)	0.694 (0.020)	0.775 (0.021)	0.810 (0.019)
0.01	0.712 (0.027)	0.727 (0.018)	0.736 (0.019)	0.818 (0.024)	0.829 (0.025)	0.815 (0.026)
0.1	0.711 (0.019)	0.727 (0.019)	0.748 (0.019)	0.827 (0.020)	0.830 (0.024)	0.821 (0.039)
1	0.711 (0.020)	0.729 (0.019)	0.749 (0.019)	0.828 (0.021)	0.830 (0.024)	0.824 (0.024)

TABLE VIII

AVERAGED AUCs WITH STANDARD DEVIATIONS (PARENTHESIS) OF F-SPLR AND U-SPLR WITH LAPLACE DLPA ON PhysioNet

	F-SPLR			U-SPLR		
	λ			λ		
	10	100	500	10	100	500
ϵ \ T	300	100	50	50		
0.001	0.619 (0.024)	0.695 (0.022)	0.727 (0.021)	0.656 (0.052)	0.749 (0.024)	0.798 (0.041)
0.01	0.710 (0.044)	0.727 (0.021)	0.724 (0.020)	0.810 (0.030)	0.830 (0.023)	0.824 (0.025)
0.1	0.711 (0.021)	0.727 (0.019)	0.749 (0.019)	0.828 (0.028)	0.830 (0.024)	0.825 (0.025)
1	0.711 (0.020)	0.730 (0.019)	0.749 (0.019)	0.829 (0.017)	0.830 (0.024)	0.825 (0.025)

TABLE IX

AVERAGED AUCs WITH STANDARD DEVIATIONS (PARENTHESIS) OF F-SPLR AND U-SPLR WITH GAUSS DLPA ON DIABETES

	F-SPLR			U-SPLR		
	λ			λ		
	10	100	500	10	100	500
ϵ \ T	300	100	50	50		
0.001	0.503 (0.018)	0.499 (0.019)	0.508 (0.025)	0.506 (0.025)	0.499 (0.025)	0.499 (0.018)
0.01	0.504 (0.018)	0.509 (0.009)	0.543 (0.013)	0.540 (0.012)	0.536 (0.009)	0.531 (0.013)
0.1	0.528 (0.021)	0.612 (0.006)	0.628 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
1	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)

APPENDIX C
EXPERIMENTAL RESULTS OF F-SPLR

Table XII provides additional experimental results of F-SPLR when differentiating the number of iterations of the algorithm.

TABLE X

AVERAGED AUCs WITH STANDARD DEVIATIONS (PARENTHESIS) OF F-SPLR AND U-SPLR WITH GAMMA DLPA ON DIABETES

	F-SPLR			U-SPLR		
	λ			λ		
	10	100	500	10	100	500
ϵ \ T	300	100	50	50		
0.001	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
0.01	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
0.1	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
1	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)

TABLE XI

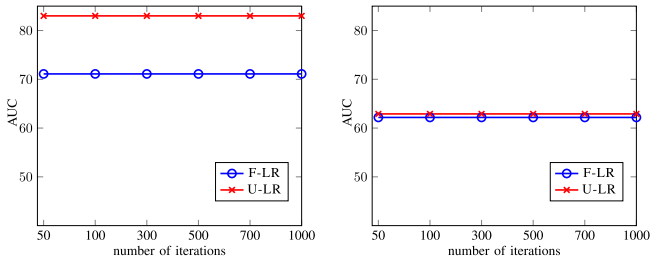
AVERAGED AUCs WITH STANDARD DEVIATIONS (PARENTHESIS) OF F-SPLR AND U-SPLR WITH LAPLACE DLPA ON DIABETES

	F-SPLR			U-SPLR		
	λ			λ		
	10	100	500	10	100	500
ϵ \ T	300	100	50	50		
0.001	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
0.01	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
0.1	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)
1	0.606 (0.006)	0.612 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)	0.629 (0.006)

TABLE XII

RESULTS OF F-SPLR AT VARIOUS ITERATION NUMBERS

Dataset	λ	T	Local	Server	CSP
PhysioNet	100	100	34.897s (0.332)	12.269s (0.177)	0.010s ($<$ 0.001)
	10	300	103.587s (0.792)	35.137s (0.424)	0.030s (0.001)
Diabetes	100	100	60.009s (0.424)	12.960s (0.192)	0.010s ($<$ 0.001)
	10	300	178.104s (1.047)	36.993s (0.515)	0.030s (0.001)

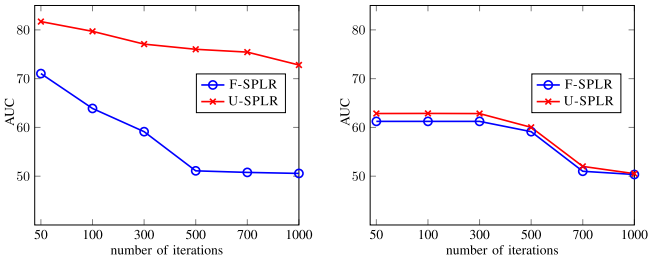


(a) PhysioNet (b) Diabetes

Fig. 5. Global AUCs of F-LR and U-LR with $\lambda = 100$ in (a) PhysioNet and (b) Diabetes datasets, depending on the number of iterations.

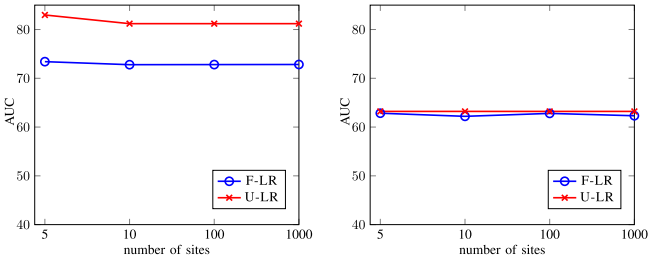
APPENDIX D
EXPERIMENTAL RESULTS OF PARAMETERS T AND K

Additional experimental results are described when differentiating the number of iterations from 50 to 1000



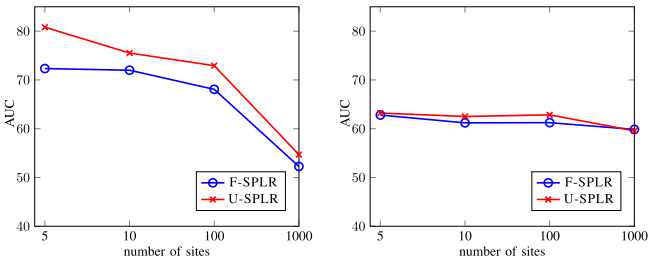
(a) PhysioNet (b) Diabetes

Fig. 6. Average AUCs of F-SPLR and U-SPLR with Gamma DLPA and $\lambda = 100$ for the privacy budgets in (a) PhysioNet and (b) Diabetes datasets, depending on the number of iterations.



(a) PhysioNet (b) Diabetes

Fig. 7. Global AUCs of F-LR and U-LR with $\lambda = 100$ in (a) PhysioNet and (b) Diabetes datasets, depending on the number of sites.



(a) PhysioNet (b) Diabetes

Fig. 8. Average AUCs of F-SPLR and U-SPLR with Gamma DLPA and $\lambda = 100$ for the privacy budgets in (a) PhysioNet and (b) Diabetes datasets, depending on the number of sites.

(Fig. 5 and 6), the number of sites from 5 to 1000 (Fig. 7 and 8) in our algorithms with privacy budgets from 0.001 to 1.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their helpful comments and remarks.

REFERENCES

[1] C. A. McCarty *et al.*, "The eMERGE Network: A consortium of biorepositories linked to electronic medical records data for conducting genomic studies," *BMC Med. Genomics*, vol. 4, no. 1, p. 13, Jan. 2011.

[2] L. Ohno-Machado *et al.*, "pSCANNER: Patient-centered scalable national network for effectiveness research," *J. Amer. Med. Inform. Assoc.*, vol. 21, no. 4, pp. 621–626, 2014.

[3] L. M. Schilling *et al.*, "Scalable architecture for federated translational inquiries network (SAFTINet) technology infrastructure for a distributed data network," *EGEMS*, vol. 1, no. 1, pp. 1–13, 2013.

[4] M. Naveed *et al.*, "Privacy and security in the genomic era," *ACM Comput. Surv.*, vol. 48, no. 1, May 2015, Art. no. 6.

[5] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[6] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong, "Publishing set-valued data via differential privacy," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 1087–1098, 2011.

[7] F. K. Dankar and K. El Emam, "The application of differential privacy to health data," in *Proc. Joint EDBT/ICDT Workshops*, 2012, pp. 158–166.

[8] D. Vu and A. Slavkovic, "Differential privacy for clinical trial data: Preliminary evaluations," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 138–143.

[9] F. Yu and Z. Ji, "Scalable privacy-preserving data sharing methodology for genome-wide association studies: An application to iDASH healthcare privacy protection challenge," *BMC Med. Inform. Decis. Making*, vol. 14, p. S3, Dec. 2014.

[10] N. Mohammed, R. Chen, B. C. M. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2011, pp. 493–501.

[11] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 289–296.

[12] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[13] D. Kifer, A. Smith, and A. Thakurta, "Private convex empirical risk minimization and high-dimensional regression," in *Proc. Conf. Learn. Theory*, 2012, pp. 1–25.

[14] J. Awan and A. Slavković, "Structure and sensitivity in differential privacy: Comparing k-norm mechanisms," 2018, *arXiv:1801.09236*. [Online]. Available: https://arxiv.org/abs/1801.09236

[15] F. Yu, M. Rybar, C. Uhler, and S. E. Fienberg, "Differentially-private logistic regression for detecting multiple-SNP association in GWAS databases," in *Privacy in Statistical Databases*. Cham, Switzerland: Springer, 2014, pp. 170–184.

[16] Z. Ji, X. Jiang, S. Wang, L. Xiong, and L. Ohno-Machado, "Differentially private distributed logistic regression using private and public data," *BMC Med. Genomics*, vol. 7, no. 1, p. S14, May 2014.

[17] I. Hegedus and M. Jelasity, "Distributed differentially private stochastic gradient descent: An empirical study," in *Proc. 24th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Feb. 2016, pp. 566–573.

[18] S. E. Fienberg, W. J. Fulp, A. B. Slavkovic, and T. A. Wrobel, "'Secure' log-linear and logistic regression analysis of distributed databases," in *Proc. Int. Conf. Privacy Stat. Databases*. Berlin, Germany: Springer, 2006, pp. 277–290.

[19] S. E. Fienberg, Y. Nardi, and A. B. Slavković, "Valid statistical analysis for logistic regression with multiple sources," in *Proc. Annu. Workshop Inf. Privacy Nat. Secur.* Berlin, Germany: Springer, 2008, pp. 82–94.

[20] Y. Nardi, S. E. Fienberg, and R. J. Hall, "Achieving both valid and secure logistic regression analysis on aggregated data from different private sources," *J. Privacy Confidentiality*, vol. 4, no. 1, pp. 189–220, 2012.

[21] W. Li, H. Liu, P. Yang, and W. Xie, "Supporting regularized logistic regression privately and efficiently," *PLoS ONE*, vol. 11, no. 6, 2016, Art. no. e0156479.

[22] H. Shi *et al.*, "Secure multi-party computation grid logistic regression (SMAC-GLORE)," *BMC Med. Inform. Decis. Making*, vol. 16, no. 3, p. 89, Jul. 2016.

[23] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 19–38.

[24] J. Snoke, T. R. Brick, A. Slavković, and M. D. Hunter, "Providing accurate models across private partitioned data: Secure maximum likelihood estimation," *Ann. Appl. Statist.*, vol. 12, no. 2, pp. 877–914, 2018.

[25] M. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1876–1884.

[26] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.

[27] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6346–6357.

[28] W. Xie, Y. Wang, S. M. Boker, and D. E. Brown, "PrivLogit: Efficient privacy-preserving logistic regression by tailoring numerical optimizers," 2016, *arXiv:1611.01170*. [Online]. Available: https://arxiv.org/abs/1611.01170

[29] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.
[30] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and evaluation," *JMIR Med. Informat.*, vol. 6, no. 2, p. e19, 2018.
[31] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Med. Genomics*, vol. 11, no. 4, p. 83, 2018.
[32] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Scalable and secure logistic regression via homomorphic encryption," in *Proc. 6th ACM Conf. Data Appl. Secur. Privacy*, 2016, pp. 142–144.
[33] P. S. Kamath and W. Kim, "The model for end-stage liver disease (MELD)," *Hepatology*, vol. 45, no. 3, pp. 797–805, 2007.
[34] *Framingham Risk Functions*. Accessed: Jul. 24, 2019. [Online]. Available: https://www.framinghamheartstudy.org/fhs-risk-functions/diabetes/
[35] C. W. Hug and P. Szolovits, "ICU acuity: Real-time models versus daily models," in *Proc. AMIA Annu. Symp.*, Jan. 2009, pp. 260–264.
[36] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *Proc. IJCAI-Workshop Mach. Learn. Inf. Filtering*, vol. 1, 1999, pp. 61–67.
[37] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics) Secaucus, NJ, USA: Springer-Verlag, 2006.
[38] Y. Wu, X. Jiang, J. Kim, and L. Ohno-Machado, "Grid binary logistic regression (GLORE): Building shared models without sharing data," *J. Amer. Med. Inform. Assoc.*, vol. 19, no. 5, pp. 758–764, 2012.
[39] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Colloquium Automata, Lang. Programm. (ICALP)*, vol. 4052. Venice, Italy: Springer-Verlag, Jul. 2006, pp. 1–12.
[40] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*. Berlin, Germany: Springer, 2006, pp. 265–284.
[41] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace Distribution and Generalizations: A Revisit With Applications to Communications, Economics, Engineering, and Finance*. New York, NY, USA: Springer, 2012.
[42] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2010, pp. 735–746.
[43] G. Ács and C. Castelluccia, "I have a DREAM! (Differentially private smart metering)," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2011, pp. 118–132.
[44] S. Goryczka and L. Xiong, "A comprehensive comparison of multiparty secure additions with differential privacy," *IEEE Trans. Dependable Sec. Comput.*, vol. 14, no. 5, pp. 463–477, Oct. 2017.
[45] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proc. 3rd ACM Workshop Cloud Comput. Secur. Workshop*, 2011, pp. 113–124.
[46] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, and X. Jiang, "Privacy-preserving patient similarity learning in a federated environment: Development and analysis," *JMIR Med. Informat.*, vol. 6, no. 2, p. e20, 2018.
[47] X. Jiang, M. Kim, K. Lauter, and Y. Song, "Secure outsourced matrix computation and application to neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 1209–1222.
[48] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Adv. Cryptol. 23rd Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*. Cham, Switzerland: Springer, 2017, pp. 409–437.
[49] D. Böhning, "The lower bound method in probit regression," *Comput. Statist. Data Anal.*, vol. 30, no. 1, pp. 13–17, 1999.
[50] S. M. Marjugi and W. J. Leong, "Diagonal hessian approximation for limited memory quasi-Newton via variational principle," *J. Appl. Math.*, vol. 2013, Nov. 2013, Art. no. 523476.
[51] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulat.*, vol. 101, no. 23, pp. e215–e220, 2000.
[52] B. Strack *et al.*, "Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records," *BioMed Res. Int.*, vol. 2014, Apr. 2014, Art. no. 781670.
[53] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *J. Math. Cryptol.*, vol. 9, no. 3, pp. 169–203, 2015.
[54] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for approximate homomorphic encryption," in *Advances in Cryptology—EUROCRYPT*. Cham, Switzerland: Springer, 2018, pp. 360–384.
[55] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1656–1665.
[56] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 245–248.

Miran Kim received the Ph.D. degree in mathematical sciences from Seoul National University, Seoul, South Korea, in 2017. She was a Post-Doctoral Researcher with the Health Department of Biomedical Informatics, University of California at San Diego (UCSD), USA, from 2017 to 2018. She is currently an Assistant Professor with the School of Biomedical Informatics (SBMI), The University of Texas Health Science Center at Houston (UTHealth), Houston, TX, USA. Her research interests include cryptography, computational number theory, data privacy, and privacy-preserving machine learning.

Junghye Lee received the B.S. and Ph.D. degrees from the Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, South Korea. She was a Post-Doctoral Researcher with the Biomedical Informatics Department, University of California at San Diego (UCSD), USA. She is currently an Assistant Professor with the School of Management Engineering, Ulsan National Institution of Science and Technology, Ulsan, South Korea. Her main research interests include machine learning, statistical learning, probabilistic graphical models, and healthcare data privacy and security.

Lucila Ohno-Machado received the Medical degree from the University of São Paulo and the Ph.D. degree in medical information sciences and computer science from Stanford University. She is the Associate Dean of Informatics and Technology and the Founding Chair of the Health Department of Biomedical Informatics, University of California at San Diego (UCSD). Her research interests include biomedical informatics, predictive modeling, and biomedical data analytics. She is elected to the National Academy of Medicine, the American College of Medical Informatics, the American Institute for Medical and Biological Engineering, and the American Society for Clinical Investigation.

Xiaoqian Jiang is a Christopher Sarofim Associate Professor and the Director of the Center for Health Security and Phenotyping, SBMI, UTHealth. His research interests include predictive models in biomedicine, health privacy, and calibration. He has received twice the Distinguished Paper Award from the American Medical Informatics Association's Clinical Research Informatics Summit. He is an Associate Editor of *BMC Biomedical Informatics and Decision Making*.