IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 10, 2018, accepted August 12, 2018, date of publication August 22, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2866697

Ensemble Method for Privacy-Preserving Logistic Regression Based on Homomorphic Encryption

JUNG HEE CHEON[1], DUHYEONG KIM[1], YONGDAI KIM[2], AND YONGSOO SONG[3]

[1]Department of Mathematical Sciences, Seoul National University, Seoul 08826, South Korea
[2]Department of Statistics, Seoul National University, Seoul 08826, South Korea
[3]Department of Computer Science and Engineering, University of California at San Diego, San Diego, CA 92093 USA

Corresponding author: Duhyeong Kim (doodoo1204@snu.ac.kr)

The work of J. H. Cheon was supported by the Institute for Information & Communications Technology Promotion (IITP) Grant through the Korean Government (MSIT), (Development of lattice-based post-quantum public-key cryptographic schemes), under Grant 2017-0-00616. The work of D. Kim was supported in part by the Institute for Information & Communications Technology Promotion (IITP) Grant through the Korean Government (MSIT), (Development of lattice-based post-quantum public-key cryptographic schemes), under Grant 2017-0-00616 and in part by NRF (National Research Foundation of Korea) Grant through the Korean Government (Global Ph.D. Fellowship Program) under Grant NRF-2016H1A2A1906584. The work of Y. Kim was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) under Grant NRF-2017R1A2B2006102.

ABSTRACT Homomorphic encryption (HE) is one of promising cryptographic candidates resolving privacy issues in machine learning on sensitive data such as biomedical data and financial data. However, HE-based solutions commonly suffer from relatively high computational costs due to a large number of iterations in the optimization algorithms such as gradient descent (GD) for the learning phase. In this paper, we propose a new method called ensemble GD for logistic regression, a commonly used machine learning technique for binary classification. Our ensemble method reduces the number of iterations of GD, which results in substantial improvement on the performance of logistic regression based on HE in terms of speed and memory. The convergence of ensemble GD based on HE is guaranteed by our theoretical analysis on the erroneous variant of ensemble GD. We implemented ensemble GD for the logistic regression based on an approximate HE scheme HEAAN on MNIST data set and Credit data set from UCI machine learning repository. Compared to the standard GD for logistic regression, our ensemble method requires only about 60% number of iterations, which results in 60–70% reduction on the running time of total learning procedure in encrypted state, and 30–40% reduction on the storage of encrypted data set.

INDEX TERMS Ensemble, gradient descent with errors, homomorphic encryption, privacy-preserving logistic regression.

I. INTRODUCTION

Machine learning has received much attentions recently due to its strong ability to solve various real world problems in artificial intelligence, bioinformatics, medical sciences, marketings and so on. In particular, machine learning can extract useful information from big data without relying much on domain experts' knowledge. For various machine learning methodologies and their applications, see [26] and [40].

Among various machine techniques, the logistic regression is a popular one for classification due to that it is not only simple enough to be applied to various problems but also competitive to other nonlinear classification algorithms in prediction accuracy. Moreover, the related loss function defined as the negative log-likelihood of the logistic regression is known to have many desirable properties and is used with more complicated classification algorithms such as the gradient boosting [27]. See [23] for comparison of the logistic regression with other machine learning algorithms.

Privacy has been an important issue in machine learning. Privacy-preserving machine learning deals with hiding a person's sensitive identity without losing the usability of data. Sensitive identities include some private information about persons, companies, and governments that have to be suppressed before shared or published. Four practically used methodologies for privacy preserving in machine learning are anonymization, perturbation, randomization and condensation [38], but loss of information is indispensable for those methods. For overview of privacy-preserving methods, refer to [48].

Applying cryptographic tools enables us to a way to prevent the loss of information while preserving the privacy during machine learning. Homomorphic Encryption (HE),

which allows arithmetics over encrypted data without decryption, has gained a lot of attentions for preventing the leakage of private information such as biomedical data (e.g., genotype/phenotype) and financial data (e.g., private asset) during machine learning. Since one can compute circuits based on HE without revealing any private information in an offline stage, HE is regarded to be very appropriate cryptographic solution for privacy issues in machine learning. There also have been some researches exploiting other cryptographic tools for privacy-preserving machine learning such as multi party computation (MPC), but HE has relative advantages compared to MPC in terms of matrix and vector operations and the availability of offline-stage computations for total learning procedure.

Besides the attractive functionality of HE in privacy preservation, a bottleneck in application of HE to machine learning is relatively high computational cost. When computing a circuit for machine learning based on HE, the depth of the circuit is the most important factor determining parameters of HE, and consequently effects on the computational cost of the homomorphic evaluation of the circuit. However, most commonly used optimization algorithms of machine learning such as gradient descent (GD) and Newton-Raphson methods are iterative algorithms, and the depth of a circuit for machine learning grows linearly to the number of iterations. In other words, there exists a limitation on reducing the computational cost of machine learning based on HE without decreasing the number of iterations in the optimization algorithms for the logistic regression. From this observation, we aimed to develop a new method for machine learning starting from the logistic regression, which requires lower number of iterations compared to previous methods.

A. OUR CONTRIBUTION

We propose a new ensemble method for learning the logistic regression which is much more efficient than the standard GD, especially when it is applied with homomorphic encryption. We first define an ensemble variant of GD, called ensemble GD, as an optimization algorithm of the logistic regression. We show that our ensemble GD method requires less iterations than the standard GD method to obtain a prediction model sufficiently near to the optimal one, and the reduction on the number of iterations consequently gives a significant improvement on the performance of the logistic regression based on HE in terms of speed and memory.

The GD algorithm for training of a logistic regression model requires the evaluation of sigmoid. The sigmoid function is usually approximated by a polynomial to be efficiently computed on HE system. Due to an approximation error, we do not have the exact gradient and thus we should consider an erroneous variant of ensemble GD which we call ensemble GD with errors. We provide a theoretical result on the convergence of ensemble GD with errors, which ensures that the ensemble GD for the logistic regression based on HE still works well even if some errors from the polynomial approximation are added.

By implementing the ensemble GD for the logistic regression on some public datasets such as MNIST handwritten digit dataset [3] and Credit dataset from UCI repository [1] in unencrypted state, we experimentally show that our ensemble method requires substantially less iterations compared to the standard method to obtain a certain level of prediction accuracy measured by area under receiver operating characteristic (AUC). To measure the performance of ensemble GD for the logistic regression based on HE, we applied an approximate HE scheme HEAAN [15], which has shown best performance on learning the logistic regression based on HE [36]. For MNIST dataset, the ensemble GD for the logistic regression based on HEAAN obtains 0.983 AUC within 12 hours by running 14 iterations, while the standard GD took more than 32 hours to obtain the same AUC by running 22 iterations. Since HEAAN does not allow exact computations, the errors from approximate computations may disrupt the convergence of ensemble GD in encrypted state. However, our theoretical result on the ensemble GD with errors verifies the convergence of the ensemble GD based on HEAAN.

B. RELATED WORKS

Integrating Data for Analysis, Anonymization and SHaring (IDASH), a national center for biomedical computing in United States, has hosted a competition providing various real-world problems related to biomedical privacy since 2014, and privacy-preserving logistic regression based on HE has been adopted as one of the main tasks since last year. Many of the cryptography research teams from all over the world participated in the competition with various HE schemes and libraries [11], [12], [15], [16], [19], [25], [32], [33], [35]. Various optimization methods have been applied to the logistic regression based on HE, such as GD [10], [36], [37] and a simplified Hessian Newton Method [6], and one of the submitted solutions exploited an approximate closed-form of logistic regression [18]. A solution using GD [36] based on the approximate HE scheme HEAAN [15] showed the best performance among the submitted solutions [2]. There also have been studies on privacy-preserving logistic regression based on HE [4], [37]. In particular, [4] focused on exploiting only additive homomorphic encryption (AHE) such as Paillier encryption [44]. Some other HE-based researches evaluated the prediction phase of neural networks [31], [34], [45].

MPC is another cryptographic primitive for secure computation. It performs a secure protocol which allows several parties to compute a certain function over their inputs while keeping them private. SecureML [39] suggested privacy-preserving machine learning methods based on the two-party computation: one is a purely MPC-based solution, and the other is a hybrid solution based on MPC and AHE. The hybrid solution required much less communication overhead but took a longer time for training compared to the purely MPC-based solution. In addition, there have been a few researches which evaluates deep neural networks using

the MPC technique [39], [46]. However, MPC-based solutions commonly assume the semi-honest model that there is no collusion between parties.

There have been proposed various ensemble methods in machine learning including the logistic regression [21], [49], [50]. General ensemble methods for machine learning were introduced [21] and an ensemble method on feature selection for the logistic regression was proposed in [49], but their aim is to provide a better estimator than the maximum likelihood estimator (MLE) of the logistic regression, while our objective is to construct an algorithm finding the MLE with less iterations. In [50], they proposed a parallel stochastic GD method with both theoretical and experimental analysis on statistical errors. However, their purpose was not a privacy-preserving machine learning so that they did not consider computational bottleneck due to HE.

C. ROAD MAP

In Section II, we give some backgrounds on the logistic regression and homomorphic encryption. In Section III, we propose our new ensemble GD for the logistic regression, and present a theoretical result on the convergence of ensemble GD with errors. In Section IV, we provide experimental results on ensemble GD for the logistic regression. In Section V, we summarize our work and suggest some follow-up studies of this work.

II. BACKGROUNDS

A. LOGISTIC REGRESSION FOR BINARY CLASSIFICATION

Machine learning (ML) is a generic term to learn something from data that are assumed to be random. For a binary classification, the most common ML technique is (binary) logistic regression. Let $\mathcal{L} = \{(x_i, y_i)\}_{1\le i\le n}$ be a given dataset where $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ for $1 \le i \le n$. The goal of learning the logistic regression is to find the optimal point $\boldsymbol{\beta} \in \mathbb{R}^d$ which maximizes the likelihood function $\prod_{i=1}^n \Pr(y_i|x_i) = \prod_{i=1}^n 1/(1+\exp(-y_i x_i^T \boldsymbol{\beta}))$ where the superscript T denotes the transpose of a vector. Taking a logarithm, the equivalent goal is to find $\boldsymbol{\beta}$ which minimizes

$$C(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \phi(y_i x_i^T \boldsymbol{\beta}) \tag{1}$$

where $\phi(z) := \log(1+\exp(-z))$. We call the negative log-likelihood function C the *logistic loss function* of given dataset $\mathcal{L}$.

A popularly used optimization algorithm to minimize $C(\boldsymbol{\beta})$ is a gradient descent (GD) algorithm. Let $\nabla C(\boldsymbol{\beta})$ be the gradient vector of $C(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. Let $\boldsymbol{\beta}^{(t)}$ be the value obtained at the t-th iteration of the GD algorithm. Then, the GD algorithm updates $\boldsymbol{\beta}^{(t)}$ by

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \gamma_t \nabla C(\boldsymbol{\beta}^{(t)}), \tag{2}$$

where $\gamma_t > 0$ is a prespecified learning rate for $t \ge 0$. If $\boldsymbol{\beta}^{(t)}$'s are contained in some compact subset D of $\mathbb{R}^d$, then it is known that $\boldsymbol{\beta}^{(t)}$ converges to $\widehat{\boldsymbol{\beta}}$ as $t \to \infty$ with the convergence rate $O(\rho^t)$ for some $0 < \rho < 1$ [42], [43] since the logistic loss function is strongly convex on D provided the design matrix $(x_1, \ldots, x_n) \in \mathbb{R}^{d\times n}$ has a full-rank. Note that $\nabla C(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^n \phi^{(1)}(y_i x_i^T \boldsymbol{\beta}) \cdot y_i x_i$ where $\phi^{(1)}(z) = -\exp(-z)/(1+\exp(-z)) = -S(-z)$ for the sigmoid function $S(z) := 1/(1+\exp(-z))$. Therefore, Equation (2) is expressed as

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \frac{\gamma_t}{n}\sum_{i=1}^{n} S(-z_i^T \boldsymbol{\beta}^{(t)}) \cdot z_i \tag{3}$$

where $z_i := y_i \boldsymbol{\beta}_i$ for $1 \le i \le n$. We denote $\boldsymbol{\beta}^{(t)}$ after sufficient number of iterations by a prediction model of logistic regression.

B. HOMOMORPHIC ENCRYPTION

Homomorphic Encryption is a cryptosystem which allows arithmetic operations such as an addition and a multiplication over encrypted data without decryption process. From this attractive property which has not been achieved in any other cryptosystems, HE is regarded to be a promising solution which prevents private information leakage during analyses on sensitive data such as biomedical data and financial data. A number of HE schemes [7]–[9], [15]–[17], [22], [24], [25], [29], [30] have been suggested following Gentry's blueprint [28], and researches on various real-world applications of HE such as the logistic regression [6], [18], [36], [37], prediction phase of deep neural network [31], [34], cyber physical system [14] have been progressed until recently.

An HE scheme basically consists of key generation, encryption, decryption, and homomorphic evaluation (addition and multiplication) algorithms:

- $\mathtt{Setup}(\lambda, L)$. For inputs security parameter λ and level parameter L, output the parameters params of the given HE scheme determined by λ and L.
- $\mathtt{KeyGen}(\mathsf{params})$. Output the secret key sk, the public key pk, and the (public) evaluation key evk.
- $\mathtt{Enc}_{\mathsf{pk}}(\mathfrak{m})$. For an input plaintext $\mathfrak{m}$, output an encryption ct of $\mathfrak{m}$, *i.e.*, $\mathtt{Dec}_{\mathsf{sk}}(\mathsf{ct}) = \mathfrak{m}$.
- $\mathtt{Dec}_{\mathsf{sk}}(\mathsf{ct})$. For an input ciphertext ct, output the decryption $\mathfrak{m}$ of ct.
- $\mathtt{Add}_{\mathsf{evk}}(\mathsf{ct}, \mathsf{ct}')$. For encryptions $\mathsf{ct}, \mathsf{ct}'$ of $\mathfrak{m}, \mathfrak{m}'$, output an encryption $\mathsf{ct}_{\mathtt{add}}$ of $\mathfrak{m}+\mathfrak{m}'$.
- $\mathtt{Mult}_{\mathsf{evk}}(\mathsf{ct}, \mathsf{ct}')$. For encryptions $\mathsf{ct}, \mathsf{ct}'$ of $\mathfrak{m}, \mathfrak{m}'$, output an encryption $\mathsf{ct}_{\mathtt{mult}}$ of $\mathfrak{m}\cdot\mathfrak{m}'$.

Note the above HE scheme only supports homomorphic evaluations of an L-depth circuit. In addition, the level parameter L is the most significant factor on the performance of HE, including the speed of algorithms and the size of public key and a ciphertext, since it determines the size of HE parameters.

Most of the HE schemes allow exact computations. In contrary, an HE scheme HEAAN [15] allows an approximate computations of real numbers. HEAAN is known to be perfectly fit in real-world applications since most computations in real-world applications are approximate computations.

By abandoning exact computations, HEAAN obtains a lot of advantages in ciphertext/plaintext ratio and speed of algorithms.

C. GRADIENT DESCENT WITH ERRORS

After the IDASH 2017 competition on the privacy-preserving logistic regression for biomedical data, HE has been recognized as a very attractive cryptographic primitive for privacy-preserving machine learning including the logistic regression. However, convergence theorems on the standard GD cannot be applied to interpret the nice performance of machine learning based on HE. For computational efficiency of HE, an approximate polynomial is usually exploited instead of the sigmoid function $S(x)$ in case of the logistic regression [36], which occurs an error in each iteration. Furthermore, in case of HEAAN which supports approximate arithmetics, not exact computations, a small error is added in every arithmetic operation. As a result, when running GD based on HE, one should consider the *erroneous* GD as following:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \gamma_t \nabla C(\boldsymbol{\beta}^{(t)}) + e^{(t)} \tag{4}$$

where C is a loss function and $e^{(t)}$ is a small error added during the t-th iteration. We will call this erroneous variant of GD by *GD with errors*.

Theoretical results on the convergence of GD with errors have been proposed in the field of study on low-precision (stochastic) gradient descent. In low-precision GD, only few most significant bits of $\boldsymbol{\beta}^{(t)}$ are stored in the memory during each GD iteration, so the discarded least significant bits actually represent an error $e^{(t)}$ in Equation (4). In 2015, De Sa *et al.* [20] proposed theoretical results on the convergence of erroneous stochastic gradient descent (SGD) with martingale-based analysis. They showed that the probability that the estimation in the t-th iteration is not contained in a small neighborhood of the optimal point $\widehat{\boldsymbol{\beta}}$ is $O(1/t)$. Recently, Song*et al.* [47] proposed similar theoretical results on convergence rate of GD with errors with a different approach. They showed that, when $\boldsymbol{\beta}^{(t)}$ is updated by GD with errors, $\boldsymbol{\beta}^{(t)}$ converges into some sufficiently small neighborhood of the optimal point $\widehat{\boldsymbol{\beta}}$. Let D be a sufficiently large compact subset of $\mathbb{R}^d$ so that every $\boldsymbol{\beta}^{(t)}$ for $t \geq 0$ is contained in D. Note that the size of D will depend on the choice of the initial $\boldsymbol{\beta}^{(0)}$. When assuming the following three conditions

1) $C : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is differentiable,
2) $\nabla C : \mathbb{R}^d \to \mathbb{R}^d$ is L-Lipschitz over D for a constant $L > 0$, i.e., $\|\nabla C(\boldsymbol{\beta}) - \nabla C(\boldsymbol{\beta}')\| \leq L \cdot \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$ for all $\boldsymbol{\beta}, \boldsymbol{\beta}' \in D$, and
3) $C : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is ℓ-strongly convex over D for a constant $\ell > 0$, i.e., $\big(\nabla C(\boldsymbol{\beta}) - \nabla C(\boldsymbol{\beta}')\big) \cdot (\boldsymbol{\beta} - \boldsymbol{\beta}') \geq \ell \cdot \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|^2$ for all $\boldsymbol{\beta}, \boldsymbol{\beta}' \in D$,

then the following properties hold:

Lemma 1 ([47], Th. 2): If $\gamma_k \leq L^{-1}$ and $\|e_k\|^2 \leq \frac{1}{2}\|\gamma_k \cdot \nabla C(\boldsymbol{\beta}^{(k)})\|^2$ for $0 \leq k < t$, then it holds that $C(\boldsymbol{\beta}^{(t)}) - C(\widehat{\boldsymbol{\beta}}) \leq \frac{L}{2} \cdot \|\boldsymbol{\beta}^{(0)} - \widehat{\boldsymbol{\beta}}\|^2 \cdot \prod_{k=0}^{t-1} \left(1 - \frac{\gamma_k \ell}{2}\right)$.

Lemma 2 ([47], Th. 3): For a fixed learning rate $0 < \gamma_t = \gamma \leq L^{-1}$ and for some errors $e_t \in \mathbb{R}^d$ with a bound $\|e_t\| \leq E$, let $X = \{\boldsymbol{\beta} \in D : \|\gamma \cdot \nabla C(\boldsymbol{\beta})\|^2 \leq 2 \cdot E^2\}$ be a compact subset of $\mathbb{R}^d$. If $\boldsymbol{\beta}^{(t_0)} \in X$ for some $t_0 \geq 0$, then it is satisfied that $C(\boldsymbol{\beta}^{(t)}) \leq M$ for all $t > t_0$ where $M = \sup_{\boldsymbol{\beta} \in X}\{C(\boldsymbol{\beta})\} + \frac{1}{2\gamma}E^2$.

Remark 1: Combining Lemma 1 and Lemma 2, $\boldsymbol{\beta}^{(t)}$ converges to $\widehat{\boldsymbol{\beta}}$ with the convergence rate $O(\rho^t)$ for $0 < \rho = 1 - \frac{\gamma\ell}{2} < 1$ until entering $X = \{\boldsymbol{\beta} \in D : \|\gamma \cdot \nabla C(\boldsymbol{\beta})\|^2 \leq 2 \cdot E^2\}$, and then oscillates in $Y := \{\boldsymbol{\beta} \in D : C(\boldsymbol{\beta}) \leq M\} \supset X$. Note that both compact sets X and Y converges to a point set $\{\widehat{\boldsymbol{\beta}}\}$ as $E \to 0$. Namely, X and Y can be sufficiently small sets containing $\boldsymbol{\beta}^{(t)}$ by controlling the upper bound of errors E.

Remark 2: The fixed learning rate condition in Lemma 2 is actually not a necessary condition for Lemma 2. To be precise, the condition $0 < \gamma_t = \gamma \leq L^{-1}$ can be relaxed to $0 < \gamma \leq \gamma_t \leq L^{-1}$ for $t \geq 0$. In this case, the lemma also holds under the same definition of X.

III. ENSEMBLE APPROACH FOR PRIVACY-PRESERVING LOGISTIC REGRESSION

In this section, we propose the ensemble GD method, and then apply it to the logistic regression. We claim that the ensemble method perfectly fits to homomorphic encryption, by showing that ensemble GD reduces the expected number of iterations compared to standard GD. The level parameter L (see Section II-B) is very crucial factor on the efficiency of homomorphic encryption since it determines the size of whole parameters of homomorphic encryption. Namely, the level parameter L significantly effects the public key size, the ciphertext size, and the speed of all algorithms. The important point is that the level parameter L linearly grows in the number of iterations, *i.e.*, $L = O(t)$ where t denotes the total number of iterations. As a result, we are able to conclude that reducing the number of iterations through ensemble GD would lead to substantial enhancement on the performance of privacy-preserving logistic regression based on homomorphic encryption.

We first present our ensemble GD algorithm and show that the expected number of iterations asymptotically decreases compared to standard GD. Then, we additionally analyze the convergence of ensemble GD when it is homomorphically computed based on an approximate HE scheme HEAAN. In the convergence analysis we consider the erroneous variant of ensemble GD, which is called *ensemble GD with errors*, and the analysis consequently justifies that an output of ensemble GD for the logistic regression based on the approximate HE scheme HEAAN is very close to the optimal point for proper settings on HEAAN parameters and polynomial approximation of sigmoid.

A. ENSEMBLE GRADIENT DESCENT

Let $\mathcal{L} = \{(\boldsymbol{\beta}_i, y_i)\}_{1 \leq i \leq n}$ be a given dataset of size n. Let $\widehat{\boldsymbol{\beta}}(\mathcal{L}, \boldsymbol{\beta}_0, t)$ denotes the estimation of $\boldsymbol{\beta}$ obtained after t

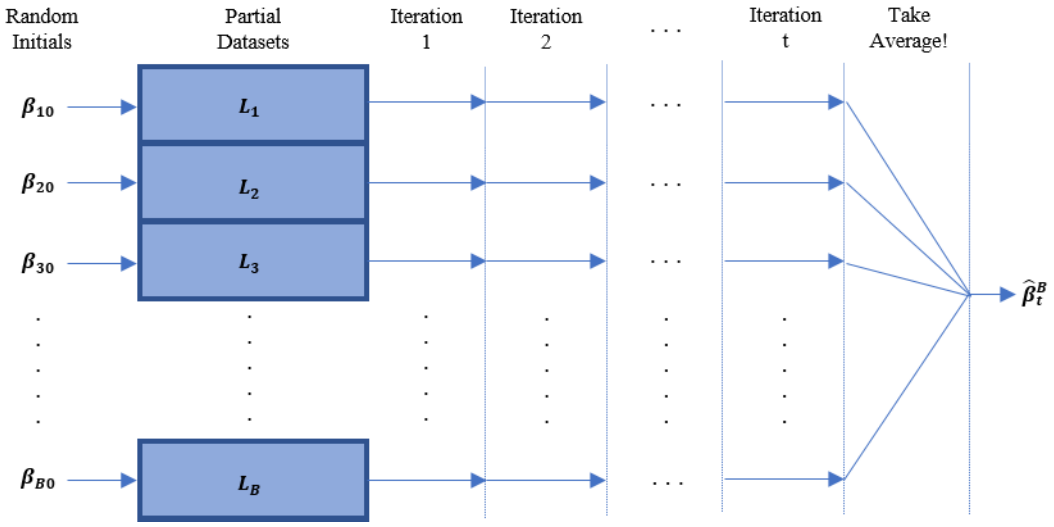

FIGURE 1. A simple description of ensemble gradient descent.

iteration of the gradient descent algorithm with the initial $\boldsymbol{\beta}_0$ and data $\mathcal{L}$. Suppose $n = Bm$ for some positive integers B and m, and then we split $\mathcal{L}$ into B many disjoint subsets $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_B$ of size m. The proposed method so-called ensemble GD is to estimate $\boldsymbol{\beta}$ by

$$\widehat{\boldsymbol{\beta}}_t^B = \frac{1}{B}\sum_{b=1}^{B} \widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t),$$

where $\boldsymbol{\beta}_{b0}$ are random initial solutions. To put it simply, ensemble GD is to run standard GD on each partial dataset $\mathcal{L}_b$ with a random initial $\boldsymbol{\beta}_{b0}$ and then take an average on resulting solutions $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ from each of the partial datasets, as described in Fig. 1. Note that those initials $\{\boldsymbol{\beta}_{b0}\}_{1\le b\le B}$ are usually randomly chosen from some bounded subset of $\mathbb{R}^d$. Our ensemble GD algorithm is described in Algorithm 1.

Algorithm 1 Ensemble Gradient Descent (Ensemble GD)

Input: Partition $\{\mathcal{L}_b\}_{1\le b\le B}$ of training data $\mathcal{L}$ with loss functions $\{C_b\}_{1\le b\le B}$, learning rates γ_k, the number of iterations t, random initials $\{\boldsymbol{\beta}_{b0}\}_{1\le b\le B}$

Output: Ensemble estimator $\widehat{\boldsymbol{\beta}}_t^B$ at t-th iteration

```
for b = 1, 2, ..., B do
    for k = 0, 1, ..., t − 1 do
        β_{b(k+1)} ← β_{bk} − γ_k ∇C_b(β_{bk})
    end
end
β̂_t^B ← (1/B) Σ_{b=1}^B β_{bt};
```

Our ensemble GD method is applicable to learning the logistic regression, *i.e.*, each loss function C_b in Algorithm 1 is the logistic loss function for a partial dataset $\mathcal{L}_b$. In this case, $C_b(\boldsymbol{\beta}) = \frac{1}{m}\sum_{(\boldsymbol{\beta}_i, y_i)\in\mathcal{L}_b} \phi(z_i^T\boldsymbol{\beta})$ and $\nabla C_b(\boldsymbol{\beta}) = -\frac{1}{m}\sum_{(\boldsymbol{\beta}_i, y_i)\in\mathcal{L}_b} S(-z_i^T\boldsymbol{\beta})\cdot z_i$ where $z_i = y_i\boldsymbol{\beta}_i$ for $1 \le i \le n$ and $|\mathcal{L}_b| = m$ for $1 \le b \le B$.

B. STATISTICAL GUARANTEE OF ENSEMBLE GRADIENT DESCENT

The motivation of ensemble GD is as follows. Let $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ be the minimizer of the empirical risk based on data $\mathcal{L}$. That is, $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ is the target we want to find. Note that $\widehat{\boldsymbol{\beta}}(\mathcal{L}) \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/n)$ for the true parameter $\boldsymbol{\beta}^*$ and some positive definite matrix Σ. Thus, it suffices to find a solution $\tilde{\boldsymbol{\beta}}$ satisfying $\tilde{\boldsymbol{\beta}} \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/n)$ so that $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ and $\tilde{\boldsymbol{\beta}}$ are asymptotically equivalent.

If loss functions corresponding to $\mathcal{L}_b$ for $1 \le b \le B$ are strongly convex and $\gamma_t \le L^{-1}$ for all $t \ge 0$, then it holds that

$$\|\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t) - \widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\| = O(\rho^t). \tag{5}$$

for some $0 < \rho < 1$ [41], [43]. It is well known that the minimizers for each of the partial data sets $\{\widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\}_{1\le b\le B}$ are independent, and follow a normal distribution

$$\widehat{\boldsymbol{\beta}}(\mathcal{L}_b) \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/m) \tag{6}$$

asymptotically for some positive definite matrix Σ. By (5) and (6), we have

$$\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \theta_{b0}, t) \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/m)$$

when $t > O(\log m)$. Thus, we expect

$$\widehat{\boldsymbol{\beta}}_t^B \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/n). \tag{7}$$

For the standard GD algorithm, we need $t > O(\log n)$ for $\widehat{\boldsymbol{\beta}}(\mathcal{L}, \boldsymbol{\beta}_0, t)$ to be around $1/\sqrt{n}$ neighborhood of $\boldsymbol{\beta}^*$, but we only need $t > O(\log m)$ for the ensemble GD algorithm.

Note that our argument is valid only when (6) is valid. Hence, m should not be too small. It is known that if

$B = o(\sqrt{n})$, the ensemble GD method is asymptotically equivalent to standard GD [5], [13], [51]. When n is not sufficiently large enough, we may use bootstrap samples for $\mathcal{L}_b$, *i.e.*, choose partial sets $\{\mathcal{L}_b\}_{1\le b\le B}$ allowing overlapping. Even though there are no theoretical guarantees for this estimator, we expect that it works quite well in practice. In this case, the number of operations for each iteration would increase compared to the case of standard GD. However, it still gives an advantage on efficiency when evaluating based on homomorphic encryption, since the total computational cost is more significantly effected by the depth parameter than the number of operations in each iteration.

Remark 3: If the initials $\{\boldsymbol{\beta}_{b0}\}_{1\le b\le B}$ are chosen in some bounded subspace of $\mathbb{R}^d$, then all the estimations $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ for $t \ge 0$ and $1 \le b \le B$ are included in some compact subset $D \in \mathbb{R}^d$. As a result, we are also able to argue (5) in case of the logistic regression since a logistic loss function is strongly convex on the compact set D.

C. ENSEMBLE GD WITH ERRORS AND ITS CONVERGENCE

As mentioned in Section II-C, a polynomial approximation of the sigmoid function $S(x)$ occurs a small error in each iteration of ensemble GD. Therefore, if we run ensemble GD described in Algorithm 1 based on HE, we should consider GD with errors instead of standard GD when updating $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ for each $1 \le b \le B$. Note that those errors can be controlled by the quality of the polynomial approximation.

In this subsection, we propose a theoretical result on the convergence rate of ensemble GD with errors. Since the size of error does not converge to 0 as iterations progress, the exact convergence of the ensemble estimation $\widehat{\boldsymbol{\beta}}_t^B$ to the optimal point $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ never happens. Therefore, our aim is to prove that $\widehat{\boldsymbol{\beta}}_t^B$ converges into a sufficiently small set containing $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ with proper convergence rate, which is the best we are able to achieve.

Using the notations in the previous subsection, ensemble GD with errors is similarly defined by

$$\widehat{\boldsymbol{\beta}}_t^B = \frac{1}{B}\sum_{b=1}^{B} \widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t),$$

where $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ is the estimation obtained after t iterations of GD with errors, instead of standard GD, with the initial value $\boldsymbol{\beta}_{b0}$ and data $\mathcal{L}_b$ for each $1 \le b \le B$. To be precise, let $C_b : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ be the loss function corresponds to the partial data set $\mathcal{L}_b$, then $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ is updated as

$$\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t+1) = \widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t) - \gamma_t \nabla C_b(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)) + e_b^{(t)}$$

where $e_b^{(t)}$ is an error vector satisfying $\|e_b^{(t)}\| \le E$ for some $E > 0$ and $\gamma_t = \gamma < L^{-1}$. Let D be a sufficiently large compact subset of $\mathbb{R}^d$ so that $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ is contained in D for all $t \ge 0$ and $1 \le b \le B$. Let us assume that each loss function $C_b : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ satisfies those three conditions

1) $C_b : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ is differentiable
2) $\nabla C_b : \mathbb{R}^d \to \mathbb{R}^d$ is L-Lipschitz over D for a constant $L > 0$
3) $C_b : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ is ℓ-strongly convex over D for a constant $\ell > 0$

so that Lemma 1 and Lemma 2 are applicable to the case $C = C_b$ for each $1 \le b \le B$. Note that a logistic loss function is strongly convex on the compact set D, so these arguments are also valid for logistic loss.

Now we analyze the convergence of ensemble GD applying some convergence theorems on GD with errors from [47] (see Section II-C). Assume that an error in each iteration of GD with errors is bounded by E, *i.e.*, $\|e_b^{(t)}\| \le E$ for any $t \ge 0$ and $1 \le b \le B$. Applying Lemma 1 for each loss function C_b and the inequality $|C_b(\boldsymbol{\beta}) - C_b(\boldsymbol{\beta}')| \ge (\ell/2)\cdot\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|^2$ which holds for any $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathbb{R}^d$, it holds that

$$\|\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t) - \widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\| < \sqrt{\frac{L}{\ell}}\left(1 - \frac{\gamma\ell}{2}\right)^{\frac{t}{2}} \|\boldsymbol{\beta}_{b0} - \widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\|$$

if $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, k) \notin X_b := \{\boldsymbol{\beta} \in D : \|\gamma \cdot \nabla C_b(\boldsymbol{\beta})\|^2 \le 2 \cdot E^2\}$ for $0 \le k < t$. Let t_{b0} be the smallest integer such that $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t_{b0})$ enters into X_b, then the above inequality holds for $0 \le t < t_{b0}$. For any $t \ge t_{b0}$, it is satisfied that $C(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)) \le M_b$ where $M_b := \sup_{\boldsymbol{\beta}\in X_b}\{C(\beta)\} + \frac{1}{2\gamma}E^2$ by Lemma 2. Since $C_b(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)) - C_b(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b)) \ge (\ell/2)\cdot\|\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t) - \widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\|^2$, we finally obtain

$$|\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t) - \widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\| \le \sqrt{\frac{2\left(M_b - C_b(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b))\right)}{\ell}},$$

which holds for $t \ge t_{b0}$.

As discussed in Remark 1, the set X_b converges to a point set $\{\widehat{\boldsymbol{\beta}}(\mathcal{L}_b)\}$ and consequently $M_b \to C_b(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b))$ as $E \to 0$. Therefore, for a sufficiently small $E > 0$, it holds that $\sqrt{2\left(M_b - C_b(\widehat{\boldsymbol{\beta}}(\mathcal{L}_b))\right)/\ell} = o(1/\sqrt{m})$ for every $1 \le b \le B$. Under this setting, we are able to argue that $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ gets close to $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b)$ within the distance $o(1/\sqrt{m})$ for every $1 \le b \le B$ if $t = O(\log m)$. By (6), we obtain

$$\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t) \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/m)$$

when $t > O(\log m)$, which consequently implies

$$\widehat{\boldsymbol{\beta}}_t^B \overset{approx}{\sim} N(\boldsymbol{\beta}^*, \Sigma/n)$$

as in the previous subsection. Therefore, ensemble GD with errors shows asymptotically same quality with ensemble GD if errors are sufficiently small. When each loss function C_b is set to be a logistic loss for $1 \le b \le B$, then the analysis can be summarized as the following statement:

Theorem 1: For a given dataset $\mathcal{L} \subset \mathbb{R}^d \times \{\pm 1\}$ of size $n = Bm$, let $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_B$ be partitioned datasets of size m, and C_b be the logistic loss function corresponding to $\mathcal{L}_b$ for $1 \le b \le B$. Assume that the initials $\{\boldsymbol{\beta}_{b0}\}_{1\le b\le B}$ are randomly chosen in some bounded subspace of $\mathbb{R}^d$ so that all the estimations $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \beta_{b0}, t)$ for $t \ge 0$ and $1 \le b \le B$ lie in some compact subset $D \subset \mathbb{R}^d$. Then, the output of ensemble

GD with errors on logistic losses $\{C_b\}_{1\le b\le B}$ for $\{\mathcal{L}_b\}_{1\le b\le B}$ is included in $O(1/\sqrt{n})$-neighborhood of $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ within $O(\log m)$ iterations when the errors in each of the iterations are initially set to be sufficiently small.

Remark 4: As stated in Remark 2, we stress that the fixed learning rate condition is not essential: the fixed learning rate condition $0 < \gamma_t = \gamma < L^{-1}$ can be substituted to $0 < \gamma \le \gamma_t < L^{-1}$ for $t \ge 0$, *i.e.*, only the lower bound γ of learning rates γ_t is required.

IV. IMPLEMENTATION

In this section, we provide experimental results on ensemble GD for the logistic regression. We first describe an approximate HE scheme HEAAN for real numbers specifically, which we applied for our experiments on ensemble GD for the logistic regression based on HE. Although there exists an additional small error in each homomorphic operations based on HEAAN since it only supports approximate computations, Theorem 1 in the Section III-C guarantees the convergence of ensemble GD even with those errors. Next we present the choice of datasets in our implementation, and the performance of ensemble GD for the logistic regression in terms of AUC, especially with a comparison to the standard GD.

A. AN APPROXIMATE HE SCHEME HEAAN

HEAAN proposed by Cheon *et al.* [15] in 2017 is a homomorphic encryption scheme which supports arithmetics of *approximate numbers* while other schemes are used for exact computations. To be precise, let ct be a HEAAN ciphertext of a message polynomial $\mathfrak{m}$. Then, the decryption process with a secret key sk is done as

$$\mathtt{Dec}_{\mathsf{sk}}(\mathsf{ct}) = \mathfrak{m} + e \approx \mathfrak{m}$$

where e is a small error attached to the message polynomial $\mathfrak{m}$. One can observe this approximate decryption as an imperfect property. However, most of the computations in real-world applications are significant digit arithmetics on real numbers, not exact arithmetics. In this sense, the approximation concept of HEAAN perfectly fits in the real world. The scheme description of HEAAN is as following:

- $\mathtt{Setup}(\lambda, p, L)$.
  - A security parameter λ, a based integer p and a level parameter L are given as input. Set $q_\ell = p^\ell$ for $\ell = 1, \ldots, L$.
  - Choose a power-of-two integer $N = N(\lambda, q_L)$, and small distributions $\chi_{\mathtt{key}}$, $\chi_{\mathtt{enc}}$, and $\chi_{\mathtt{err}}$ over the $2N$-th cyclotomic ring $R = \mathbb{Z}[X]/(X^N + 1)$.
  - Return $\mathsf{params} \leftarrow (N, \chi_{\mathtt{key}}, \chi_{\mathtt{enc}}, \chi_{\mathtt{err}})$.
- $\mathtt{KeyGen}(\mathsf{params})$.
  - Sample $s \leftarrow \chi_{\mathtt{key}}$. Set the secret key as $\mathsf{sk} \leftarrow (1, s)$.
  - Sample $a \leftarrow U(R_{q_L})$ and $e \leftarrow \chi_{\mathtt{err}}$. Set the public key as $\mathsf{pk} \leftarrow (b, a) \in R_{q_L}^2$ where $b \leftarrow -a \cdot s + e \pmod{q_L}$.
  - Sample $a' \leftarrow U(R_{q_L^2})$ and $e' \leftarrow \chi_{\mathtt{err}}$. Set the evaluation key as $\mathsf{evk} \leftarrow (b', a') \in R_{q_L^2}^2$ where $b' \leftarrow -a's + e' + q_L \cdot s^2 \pmod{q_L^2}$.
- $\mathtt{Enc}_{\mathsf{pk}}(\mathfrak{m})$. For $\mathfrak{m} \in R$, sample $v \leftarrow \chi_{\mathtt{enc}}$ and $e_0, e_1 \leftarrow \chi_{\mathtt{err}}$. Output $v \cdot \mathsf{pk} + (\mathfrak{m} + e_0, e_1) \pmod{q_L}$.
- $\mathtt{Dec}_{\mathsf{sk}}(\mathsf{ct})$. For $\mathsf{ct} = (c_0, c_1) \in R_{q_\ell}^2$, output $\mathfrak{m}' = c_0 + c_1 \cdot s \pmod{q_\ell}$.
- $\mathtt{Add}(\mathsf{ct}, \mathsf{ct}')$. For $\mathsf{ct}, \mathsf{ct}' \in R_{q_\ell}^2$, output $\mathsf{ct}_{\mathtt{add}} \leftarrow \mathsf{ct} + \mathsf{ct}' \pmod{q_\ell}$.
- $\mathtt{Sub}(\mathsf{ct}, \mathsf{ct}')$. For $\mathsf{ct}, \mathsf{ct}' \in R_{q_\ell}^2$, output $\mathsf{ct}_{\mathtt{sub}} \leftarrow \mathsf{ct} - \mathsf{ct}' \pmod{q_\ell}$.
- $\mathtt{Mult}_{\mathsf{evk}}(\mathsf{ct}, \mathsf{ct}')$. For $\mathsf{ct} = (c_0, c_1), \mathsf{ct}' = (c_0', c_1') \in \mathcal{R}_{q_\ell}^2$, let $(d_0, d_1, d_2) = (c_0c_0', c_0c_1' + c_1c_0', c_1c_1')$ $\pmod{q_\ell}$. Output $\mathsf{ct}_{\mathtt{mult}} \leftarrow (d_0, d_1) + \lfloor q_L^{-1} \cdot d_2 \cdot \mathsf{evk} \rceil \pmod{q_\ell}$.
- $\mathtt{RS}_{\ell \to \ell'}(\mathsf{ct})$. For a ciphertext $\mathsf{ct} \in R_{q_\ell}^2$ at level ℓ, output $\mathsf{ct}' \leftarrow \lfloor (q_{\ell'}/q_\ell) \cdot \mathsf{ct} \rceil \pmod{q_{\ell'}}$.

Since each element $\mathfrak{m} \in R$ is a $\mathbb{Z}_q$-coefficient polynomial, not a real number, there should be a conversion between polynomials and real-numbers for real number computations. It is well known that the ring $\mathbb{R}[X]/(X^N + 1)$ is isomorphic to $\mathbb{C}^{N/2}$ via the invertible mapping $\sigma : f(X) \mapsto f(\zeta_M^i)_{i \in T}$ where ζ_M is an M-th primitive root of unity in $\mathbb{C}$ and $T = \langle 5 \rangle$ is a proper subgroup of the multiplicative unit group $\mathbb{Z}_M^\times$ satisfying $\mathbb{Z}_M^\times / T \simeq \{1, -1\}$. Applying this isomorphism, the encoding/decoding algorithms are constructed as follows:

- $\mathtt{Ecd}(\vec{m}; \Delta)$. For a plaintext vector $\vec{m} = (m_1, \ldots, m_{N/2})$ in $\mathbb{C}^{N/2}$ and a scaling factor $\Delta > 0$, output $\mathfrak{m} \leftarrow \lfloor \sigma^{-1}(\Delta \cdot \vec{m}) \rceil \in R$ where the rounding $\lfloor \cdot \rceil$ is component-wisely done.
- $\mathtt{Dcd}(\mathfrak{m}; \Delta)$. For $\mathfrak{m} \in R$, output $\vec{m}' = \Delta^{-1} \cdot \sigma(\mathfrak{m}) \in \mathbb{C}^{N/2}$.

From the above encoding/decoding algorithms, we are able to pack $(N/2)$ complex numbers in a single polynomial. Since $\mathbb{R}$ is a subspace of $\mathbb{C}$, it is trivial that we are also able to pack $(N/2)$ real numbers. In summary, the plaintext space of HEAAN is $\mathbb{C}^{N/2}$ which is a superset of $\mathbb{R}^{N/2}$, and every element of $\mathbb{C}^{N/2}$ is encoded to a polynomial in R via $\mathtt{Ecd}(\cdot)$ before being encrypted.

B. EXPERIMENTAL SETTINGS

All experiments on our ensemble method were implemented in C++ on Linux with Intel Xeon CPU E5-2620 v4 at 2.10GHz processor, and we used 8 threads for the acceleration of our experiments. The dataset choice and parameter selection are as follows.

1) DATASET CHOICE

On experiments of ensemble GD for the logistic regression, we chose MNIST dataset [3] from the MNIST database of handwritten digits, which is one of the most commonly used dataset for machine learning experiments, and Credit dataset [1] from UCI machine learning repository. The original MNIST dataset contains 10 classes from the number 1 to 10, so we extracted partial data corresponding to the

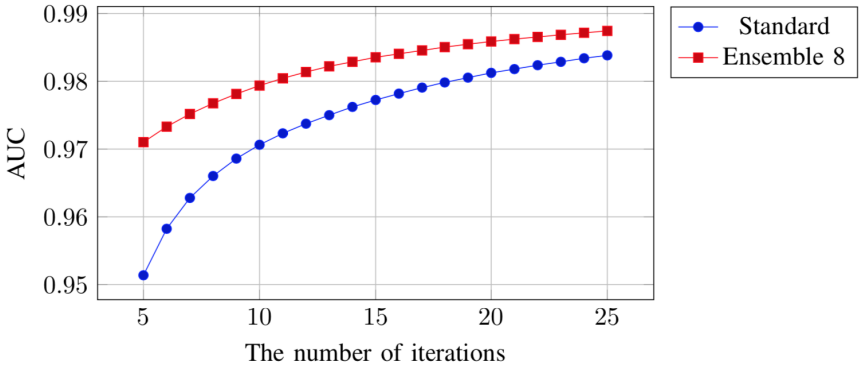

FIGURE 2. An Iteration-AUC graph of the standard/ensemble methods for MNIST dataset.

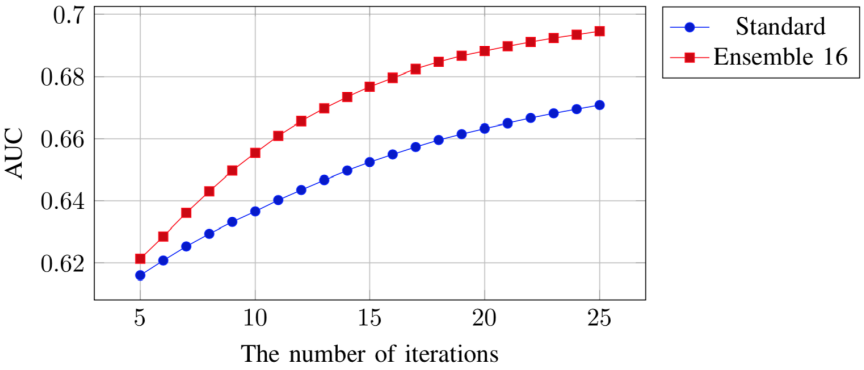

FIGURE 3. An Iteration-AUC graph of the standard/ensemble methods for Credit dataset.

number 3 and 8 for a binary classification. In addition, we compressed the number of features from 28×28 to 14×14 by taking an average for each 4-feature partition. Consequently, the MNIST dataset consists of 11,982 samples for training and 1,984 samples for test with 196 features. The Credit dataset, which contains exactly two classes, consists of 28,000 samples for training and 2,000 samples for test with 23 features.

2) PARAMETERS OF HEAAN

We applied an approximate HE scheme HEAAN to ensemble GD for the logistic regression. Following the parameter selection strategy of [36] and [37], we first choose the total number of iterations t for ensemble GD and set input parameters $p = 2$, $\lambda = 80$, and $L = 163 \cdot t + 35$. The rescaling algorithm $\mathtt{RS}_{\ell \to \ell'}(\cdot)$ is repeated for 6 times in each iteration of ensemble GD, and the bit length of a modulus is reduced for 163 bits after each iteration. The dimension N of the cyclotomic ring R was chosen to be the smallest power-of-two integer satisfying $N \geq \frac{\lambda+110}{7.2} \cdot \log q_L$ considering all known attacks. The distributions $\chi_{\mathtt{key}}$, $\chi_{\mathtt{err}}$ and $\chi_{\mathtt{enc}}$ are equally set as presented in the original paper [15] of HEAAN. A scaling factor which determines the number of precision bits was set to be $\Delta = 30$.

3) PARAMETERS OF THE ENSEMBLE GD FOR THE LOGISTIC REGRESSION

To accelerate the gradient descent process, we applied Nesterov's acceleration algorithm [41] and set learning rates to be $\gamma_k \approx 10/(k+1)$ as suggested in [36]. There exists two additional factors we should consider for ensemble GD: the number of partitions B, and the ℓ_∞-norm bound of random initials τ, *i.e.*, an absolute value of each component of $\boldsymbol{\beta}_{b0}$ for $1 \leq b \leq B$ is bounded by τ. For the MNIST dataset (resp. Credit dataset), we set $\tau = 1$ (resp. $\tau = 2$). On this setting, the optimal point $\widehat{\boldsymbol{\beta}}(\mathcal{L})$ is contained in $\{\boldsymbol{\beta} \in \mathbb{R}^d : \|\boldsymbol{\beta}\|_\infty \leq \tau\}$ for each dataset.

C. EXPERIMENTAL RESULTS

1) IMPACT OF THE ENSEMBLE METHOD.

To analyze an impact of the ensemble GD method for the logistic regression accurately, we first compare the performance of ensemble GD for the logistic regression to that of standard GD in unencrypted state. As stated in Section III-B, the theoretical analysis on the ensemble method does not hold if the number of partitions is too large so that the number of samples in each partial dataset gets too small. We experimentally compared the performance of ensemble GD for the logistic regression for various number of partitions, and finally chose the number of partitions to be 8 and 16 for MNIST and Credit datasets, respectively, which gave the best results in practice. Figure 2 and Figure 3 represent graphs of AUC with respect to the number of iterations for MNIST and Credit datasets, respectively.

The terms "Ensemble 8" and "Ensemble 16" denote the ensemble GD for the logistic regression with 8 partitions and 16 partitions, respectively, and "Standard" denotes the

TABLE 1. Performance of ensemble GD for the logistic regression based on HEAAN for MNIST and Credit datasets, and comparison with the standard GD.

Dataset	# of Samples	# of Features	GD Method	# of Iterations	Enc (sec)	Learn (min)	Storage (GB)	AUC
MNIST	11982	196	Standard	22	288	1946	5.93	0.983
			Ensemble 8	**14**	**187**	**712**	**4.25**	**0.983**
Credit	28000	24	Standard	16	45	221	1.04	0.652
			Ensemble 16	**10**	**30**	**67**	**0.65**	**0.655**

standard GD for the logistic regression. Since initials are chosen randomly, there exists a variance on AUC of each experiment. Consequently, we ran the experiment for 20 times and recorded the median of them in the graphs.

From the Iteration-AUC graphs, we can check that the ensemble method enables to obtain certain AUC with much smaller number of iterations compared to the standard method. For MNIST dataset (resp. Credit dataset), the ensemble GD with 8 partitions (resp. 16 partitions) obtains 0.98 AUC (resp. 0.66 AUC) within 11 iterations, but the standard GD requires 19 iterations to get the same AUC.

2) PERFORMANCE OF ENSEMBLE GD FOR THE LOGISTIC REGRESSION BASED ON HEAAN.

We now present the performance of ensemble GD for the logistic regression based on an approximate HE scheme HEAAN. We used HEML library [35], an HE library for the logistic regression based on HEAAN, which follows the methodology of [36] for packing a dataset into ciphertexts and computing GD algorithm based on HEAAN. For each partial dataset $\mathcal{L}_b$ for $1 \le b \le B$, we ran the same HE algorithm for GD with [36] with a random initial $\boldsymbol{\beta}_{b0}$ for t iterations, and then took an average on outputs, the encryptions of $\widehat{\boldsymbol{\beta}}(\mathcal{L}_b, \boldsymbol{\beta}_{b0}, t)$ for $1 \le b \le B$, in the last step. Consequently, a prediction model we aim to obtain is outputted in encrypted state.

The previous experiments were performed in unencrypted state, and the exact sigmoid function $S(x)$ was exploited for the logistic regression. In contrary, we substitute the sigmoid function by an approximate polynomial when applying HE on learning the logistic regression for computational efficiency. For a polynomial approximation of sigmoid, we followed a methodology of [36] and [37] using the Least Square Approximation (LSA) method on a certain interval. For Credit dataset we used a degree-5 approximate polynomial $g_5(x) = 0.5 - 0.19131\ x + 0.0045963x^3 - 0.0000412332\ x^5$ of sigmoid on the interval $[-8, 8]$ which was initially proposed in [37]. Since the number of features of MNIST dataset is much larger than that of Credit dataset, we need a larger interval than $[-8, 8]$ for an LSA polynomial approximation: a degree-5 approximate polynomial $h_5(x) = 0.5 - 0.1167694\ x + 0.0008352x^3 - 0.0000021\ x^5$ of sigmoid on the interval $[-20, 20]$ was exploited for Credit dataset. The following table shows the performance of ensemble/standard GD for the logistic regression based on HEAAN for MNIST and Credit datasets. Enc and Learn denote the running time of encrypting a dataset and total learning procedure, respectively. Storage denotes the size of an encrypted dataset.

From Table 1, we can check that our ensemble method shows a much better performance with respect to AUC compared to the standard method. Ensemble GD for the logistic regression based on HEAAN with 16 partitions obtains 0.983 AUC with 14 iterations, and the learning phase takes about 12 hours. However, the standard GD for the logistic regression based on HEAAN obtains 0.982 AUC with 22 iterations, which takes a much longer time for the learning phase. In case of Credit dataset we can also check that ensemble GD based on HEAAN shows much better performance than standard GD based on HEAAN with respect to both AUC and the running time of total learning procedure.

Note that AUCs recorded in Table 1 are a bit lower than those recorded in Iteration-AUC graphs above. As stated in Section III-C, since the sigmoid function is substituted by approximate polynomials and HEAAN supports approximate computations, we should consider the ensemble GD with errors, not the standard ensemble GD. We can expect that these errors occurred the decrease of AUC.

V. CONCLUSIONS

In this paper, we proposed an ensemble GD method for the logistic regression, which reduces the expected number of GD iterations compared to the standard GD. Since the level parameter of HE linearly grows to the number of iterations, our ensemble method consequently derived substantial reduction in the running time of total learning procedure in encrypted state and the storage of encrypted data. Since a polynomial approximation on the sigmoid function and approximate computations of HEAAN occurs a small error in each iteration, we made a new theoretical analysis on the convergence of ensemble GD with errors in section III-C, which guarantees the performance of our ensemble logistic regresion based on HE. In section IV, we also provided experimental results of ensemble GD for the logistic regression (based on HEAAN) with comparison to standard GD for public datasets. The implementation results showed our ensemble method requires much less iterations (thus requires much less running time of total learning procedure and storage of encrypted datasets).

To show the efficiency of ensemble GD, we chose a machine learning technique as the logistic regression and an HE scheme as HEAAN. However, the ensemble method is a general method: it is valid to not only the logistic loss function but also any other loss functions which are strongly convex on some compact domain, and any other HE schemes are applicable. Theorem 1 on the convergence of GD with errors in Section III-C, in fact, does not exactly fit into our implementation since we additionally applied Nesterov's acceleration algorithm. Generalization of Theorem 1 to the case of Nesterov's accelerated GD would be an important following work for our research. In addition, we experimentally chose the optimal number of partitions for MNIST and Credit datasets in this paper. Finding the strategy to choose the optimal number of partitions in the ensemble method would be very interesting following-up study on both statistics and machine learning.

VI. ACKNOWLEDGMENT

(Duhyeong Kim, Jung Hee Cheon, Yongdai Kim, and Yongsoo Song contributed equally to this work.)

REFERENCES

[1] *Default of Credit Card Clients Data Set*. Accessed: Jun. 11, 2018. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[2] *IDASH 2017 Competition*. Accessed: May 1, 2018. [Online]. Available: http://www.humangenomeprivacy.org/2017/

[3] *THE MNIST DATABASE of Handwritten Digits*. Accessed: May 28, 2018. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[4] 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.

[5] H. Battey, J. Fan, H. Liu, J. Lu, and Z. Zhu. (2015). "Distributed estimation and inference with statistical guarantees." [Online]. Available: https://arxiv.org/abs/1509.05457

[6] C. Bonte and F. Vercauteren, "Privacy-preserving logistic regression training," Cryptol. ePrint Arch., Tech. Rep. 2018/233, 2018. [Online]. Available: https://eprint.iacr.org/2018/233

[7] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *Proc. IMA Int. Conf. Cryptogr. Coding*. New York, NY, USA: Springer, 2013, pp. 45–64.

[8] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Proc. Annual Cryptology Conference*, R. Safavi-Naini and R. Canetti, Eds. New York, NY, USA: Springer, 2012, pp. 868–886.

[9] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. ITCS*, 2012, pp. 309–325.

[10] H. Chen *et al.*, "Logistic regression over encrypted data from fully homomorphic encryption," Cryptol. ePrint Arch., Tech. Rep. 2018/462, 2018. [Online]. Available: https://eprint.iacr.org/2018/462

[11] H. Chen, K. Han, Z. Huang, A. Jalali, and K. Laine. (2016). *SEAL Library*. [Online]. Available: https://www.microsoft.com/en-us/research/project/simple-encrypted-arithmetic-library/

[12] H. Chen, K. Han, Z. Huang, A. Jalali, and K. Laine. (2017). *Simple Encrypted Arithmetic Library v2.3.0*. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/12/sealmanual.pdf

[13] X. Chen and M.-G. Xie, "A split-and-conquer approach for analysis of extraordinarily large data," *Statistica Sinica*, vol. 24, no. 4, pp. 1655–1684, 2014.

[14] J. H. Cheon *et al.*, "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption," *IEEE Access*, vol. 6, pp. 24325–24339, 2018.

[15] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* New York, NY, USA: Springer, 2017, pp. 409–437.

[16] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.* New York, NY, USA: Springer, 2016, pp. 3–33.

[17] A. Costache and N. P. Smart, "Which ring based somewhat homomorphic encryption scheme is best?" in *Cryptographers' Track RSA Conf.* New York, NY, USA: Springer, 2016, pp. 325–340.

[18] J. L. Crawford, C. Gentry, S. Halevi, D. Platt, and V. Shoup, "Doing real work with fhe: The case of logistic regression," Cryptol. ePrint Arch., Tech. Rep. 2018/202, 2018. [Online]. Available: https://eprint.iacr.org/2018/202

[19] CryptoExperts. (2016). *FV-NFL Library*. [Online]. Available: https://github.com/CryptoExperts/FV-NFLlib

[20] C. M. De Sa, C. Zhang, K. Olukotun, and C. Ré, "Taming the wild: A unified analysis of hogwild-style algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2674–2682.

[21] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.* New York, NY, USA: Springer, 2000, pp. 1–15.

[22] M. V. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* New York, NY, USA: Springer, 2010, pp. 24–43.

[23] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Inform.*, vol. 35, nos. 5–6, pp. 352–359, 2002.

[24] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* New York, NY, USA: Springer, 2015, pp. 617–640.

[25] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," IACR Cryptol. ePrint Arch., Tech. Rep. 2012/144, 2012.

[26] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer series in statistics), vol. 1. New York, NY, USA: Springer-Verlag, 2001.

[27] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[28] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2009. [Online]. Available: http://crypto.stanford.edu/craig

[29] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Proc. Annu. Cryptol. Conf.* New York, NY, USA: Springer, 2012, pp. 850–867.

[30] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. Annu. Cryptol. Conf.* New York, NY, USA: Springer, 2013, pp. 75–92.

[31] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1–10.

[32] S. Halevi and V. Shoup, "Design and implementation of a homomorphic-encryption library," IBM Res., New York, NY, USA, Tech. Rep., 2013.

[33] S. Halevi and V. Shoup. (2014). *HElib Library*. [Online]. Available: https://github.com/shaih/HElib

[34] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. (2018). "Gazelle: A low latency framework for secure neural network inference." [Online]. Available: https://arxiv.org/abs/1801.05507

[35] A. Kim. (2017). *HEML Library*. [Online]. Available: https://github.com/kimandrik/HEML

[36] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," Cryptol. ePrint Arch., Tech. Rep. 2018/254, 2018. [Online]. Available: https://eprint.iacr.org/2018/254

[37] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and evaluation," *JMIR Med. Inform.*, vol. 6, no. 2, p. e19, 2018.

[38] M. B. Malik, M. A. Ghazi, and R. Ali, "Privacy preserving data mining techniques: Current scenario and future prospects," in *Proc. 3rd Int. Conf. Comput. Commun. Technol. (ICCCT)*, Nov. 2012, pp. 26–32.

[39] 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.

[40] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012, pp. 1–21.
[41] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, 1983.
[42] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. New York, NY, USA: Springer, 2004.
[43] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, vol. 30. Philadelphia, PA, USA: SIAM, 1970.
[44] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* New York, NY, USA: Springer, 1999, pp. 223–238.
[45] 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.
[46] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar. (2018). "Chameleon: A hybrid secure computation framework for machine learning applications." [Online]. Available: https://arxiv.org/abs/1801.03239
[47] Y. Song, D. Kim, and S. Gao, "A note on the convergence analysis of low-precision stochastic gradient descent," Tech. Rep., 2018.
[48] K. J. Vinoth and V. Santhi, "A brief survey on privacy preserving techniques in data mining," *IOSR J. Comput. Eng.*, vol. 18, no. 4, pp. 47–51, Jul./Aug. 2016.
[49] R. Zakharov and P. Dupont, "Ensemble logistic regression for feature selection," in *Proc. IAPR Int. Conf. Pattern Recognit. Bioinf.* Springer, 2011, pp. 133–144.
[50] J. Zhang, C. De Sa, I. Mitliagkas, and C. Ré. (2016). "Parallel SGD: When does averaging help?" [Online]. Available: https://arxiv.org/abs/1606.07365
[51] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression," in *Proc. Conf. Learn. Theory*, 2013, pp. 592–617.

JUNG HEE CHEON received the B.S. and Ph.D. degrees in mathematics from KAIST in 1991, and 1997, respectively. He is currently a Professor with the Department of Mathematical Sciences and the Director of the Cryptographic Hard problems Research Initiatives at Seoul National University (SNU). Before joining SNU, he was with ETRI, Brown University, and ICU. He received the Best Paper Award in Asiacrypt 2008 and Eurocrypt 2015. His research focuses on computational number theory, cryptology and their applications to practical problems. He was the PC Co-Chair of ANTS-XI and Asiacrypt 2015/2016. He is an Associate Editor of DCC and JCN, and served as program committee members for Crypto, Eurocrypt, and Asiacrypt.

DUHYEONG KIM received the B.S. degree from the Department of Mathematical Sciences, Seoul National University, where he is currently pursuing the Ph.D. degree supervised by Prof. J. H. Cheon. His research interests include the complexity of lattice problems and lattice-based cryptography, including homomorphic encryption.

YONGDAI KIM received the B.S. and M.S. degrees in statistics from Seoul National University (SNU) in 1991 and 1993, respectively, and the Ph.D. degree of statistics from Ohio State University, Columbus, OH, USA, in 1997. He is currently a Professor with the Department of Statistics, Seoul National University. Before joining SNU, he was at the National Institutes of Health. His research focuses on machine learning and Bayesian statistics.

YONGSOO SONG received the Ph.D. degree in mathematical sciences from Seoul National University in 2018. He is currently a Post-Doctoral Researcher with the Department of Computer Science and Engineering, University of California at San Diego, San Diego, CA, USA. His research interests include cryptographic primitives for secure computation and their applications.