IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 12, 2019, accepted August 25, 2019, date of publication September 2, 2019, date of current version September 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938759

An Attack-Based Evaluation Method for Differentially Private Learning Against Model Inversion Attack

CHEOLHEE PARK[1], DOWON HONG[1], AND CHANGHO SEO[2]
[1]Department of Mathematics, Kongju National University, Gongju 32588, South Korea
[2]Department of Convergence Science, Kongju National University, Gongju 32588, South Korea

Corresponding author: Dowon Hong (dwhong@kongju.ac.kr)

This work was supported in part by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant 2019R1A2C1003146, and in part by the Electronics and Telecommunications Research Institute (ETRI) Grant funded by the Korean Government (Core Technology Research on Trust Data Connectome) under Grant 19ZH1200.

ABSTRACT As the amount of data and computational power explosively increase, valuable results are being created using machine learning techniques. In particular, models based on deep neural networks have shown remarkable performance in various domains. On the other hand, together with the development of neural network models, privacy concerns have been raised. Recently, as privacy breach attacks on training datasets of neural network models have been proposed, research on privacy-preserving neural networks have been conducted. Among the privacy-preserving approaches, differential privacy provides a strict privacy guarantee, and various differentially private mechanisms have been studied for neural network models. However, it is not clear how appropriate privacy parameters should be chosen, considering the model's performance and the degree of privacy guarantee. In this paper, we study how to set appropriate privacy parameters to preserve differential privacy based on the resistance to privacy breach attacks in neural networks. In particular, we focus on the model inversion attack for neural network models, and study how to apply differential privacy as a countermeasure against this attack while retaining the utility of the model. In order to quantify the resistance to the model inversion attack, we introduce a new attack performance metric, instead of a survey-based approach, by leveraging a deep learning model, and capture the relationship between attack probability and the degree of privacy guarantee.

INDEX TERMS Differential privacy, differentially private learning, model inversion attack, privacy-preserving neural network.

I. INTRODUCTION

Machine learning technologies have progressed remarkably and are increasingly getting attention. In particular, models based on deep neural networks have shown remarkable performance in various domains such as face recognition, language representation, classification, etc. (e.g., [1]–[5]). These deep learning models utilize massive amounts of data, and most of the datasets contain sensitive individual information, so there is a concern about privacy breaches.

Recently, privacy invasion attacks on the training dataset of a neural network model have been proposed. In cloud-based machine learning services, i.e., machine learning as a service (MLaaS), users send input queries and MLaaS then returns confidence values corresponding to the inputs. By exploiting such systems that return confidence values corresponding to arbitrary inputs, it was revealed that an adversary can violate the privacy of training data [6]–[9]. Although basic countermeasures against privacy invasion attacks have been proposed in the MLaaS environment, they cannot fundamentally prevent such attacks.

Differential privacy [10], [11] is a rigorous notion of privacy, and research on differentially private neural network models has been actively conducted. Since differentially private mechanisms return plausible noisy results that make no statistical difference regardless of the presence or absence of any single entry in a dataset, it provides strong privacy for the entire dataset. However, it is not clear how to set a privacy

The associate editor coordinating the review of this article and approving it for publication was Longxiang Gao.

budget, considering the trade-off between privacy and utility. This is a well-known problem in the field of differential privacy.

In this paper, we study the application of differential privacy as a countermeasure against privacy breach attacks in a neural network model, and also how to set appropriate privacy parameters based on resistance to the attacks, with the aim of achieving differential privacy. In particular, we focus on model inversion attacks [7], which are among the most fatal privacy breach attacks. With a face recognition system, Fredrikson *et al.* [7] showed that parts of the training data used to train neural network models can be reconstructed. This means that the information of original data can be directly exposed. As a method to prevent the model inversion attack, we consider differentially private mechanisms for neural network-based learning model. In recent years, many mechanisms that make neural network models differentially private have been proposed [12]–[33]. Among them, we focus on Abadi *et al.'s* [19] basic differentially private stochastic gradient descent algorithm, which is one of the state-of-the-art results for differentially private deep learning. In [19], the authors proposed a differentially private stochastic gradient descent algorithm with a fine-tuned composition technique. They showed that differential privacy can be satisfied with moderate privacy costs in a non-convex optimization problem of a deep learning model by combining the Gaussian mechanism with random sampling. However, there is a lack of consideration as to the degree that a privacy guarantee should be assured to prevent actual privacy attacks, taking into account the trade-off between privacy and utility. Accordingly, from the perspective of differential privacy, we analyze how much of a privacy budget is required to prevent the model inversion attack without a significant degradation of utility.

In terms of analyzing the effectiveness of differentially private mechanisms against the model inversion attack, measuring tools for capturing changes in attack success rate depending on degree of privacy guarantee are essential. In [7], the authors quantified the efficacy of the attack using Amazon's Mechanical Turk surveys (they asked workers to guess the same person as the reconstructed image with several real face examples). However, since this approach requires a lot of time and resources, it is infeasible to use this survey-based quantification to evaluate the effectiveness of each private model with different degrees of privacy guarantee. To overcome this infeasibility, we introduce a new attack performance metric, instead of survey-based approach, by leveraging a deep learning model, and capture the relationship between attack probability and the degree of privacy guarantee. Furthermore, we experiment not only with the AT&T Laboratories Cambridge database of faces [34] that was experimented in the model inversion attack [7], but also with the VGGFace2 dataset [35] by varying the size of the dataset in order to demonstrate how differential privacy withstands the model inversion attack depending on the amount of training data.

The rest of this paper is organized as follows. Section 2 presents a review of the background. We describe the differentially private learning, the model inversion attack, and attack performance metric in section 3. In section 4, we describe experiments in detail and demonstrate the results and evaluations. Finally, we discuss some related issues in section 5, and conclude our work in section 6.

II. BACKGROUND

In this section, we briefly review the neural network model and give an overview of privacy invasion attacks for the neural network model, focusing on the model inversion attack. Then, we review the definition of differential privacy and differentially private mechanisms for neural network-based models.

A. NEURAL NETWORK MODEL

Neural network-based models, especially deep learning models, have outperformed compared with traditional techniques in various domains. By composing many layers of basic building blocks, they parameterize functions from inputs to outputs. In general, the goal in deep learning models is to output a model that fits a given finite dataset, and we train to optimize all parameters using a given dataset to achieve the goal.

Objective functions of deep neural networks is usually a non-convex optimization problem and difficult to optimize. To optimize the parameters, learning algorithms train to minimize the objective function $\mathcal{L}$ (generally, loss function or cost function). The output of the objective function $\mathcal{L}(\theta)$ over parameters θ takes the average over the training examples. Let the training dataset be $\{x_1, \ldots, x_N\}$, then $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, \boldsymbol{x}_i)$. In the context of optimizing a non-convex problem over a deep learning model, the mini-batch stochastic gradient descent (SGD) algorithm is the most common algorithm. In SGD, the optimizer sets mini-batch B by random sampling, computes gradient $\mathbf{g}_B$ over the mini-batch B: $\mathbf{g}_B = \frac{1}{|B|}\sum_{x\in B} \nabla_\theta \mathcal{L}(\theta, \boldsymbol{x})$, and then updates the parameters θ by descending $\mathbf{g}_B$ associated with learning rate.

B. PRIVACY INVASION ATTACKS ON MACHINE LEARNING

In recent years, many privacy invasion attacks on machine learning models have been proposed. One aspect of privacy invasion attacks is to infer or exploit statistical information. Homer *et al.* [36] showed that it is possible to infer the presence of a particular genome in a training dataset by comparing the published statistics about the dataset with the statistics of the general dataset. For particular machine learning models such as SVM and HMM models, Ateniese *et al.* [37] reported that general statistical information about the training dataset can be inferred through the parameters of the models. Calandrino *et al.* [38] showed that by capturing changes in the outputs of a collaborative recommender system, an adversary can infer the inputs that caused the changes. Another aspect is the malicious use of the machine learning model itself, and there are concerns that the privacy of training data can be directly invaded. Shokri *et al.* [8] focused on a fundamental

attack, called a membership inference attack, where an adversary can infer whether an input data was included in the training dataset of a target model for any type of machine learning model. They reported that overfitting is the main reason why the model is vulnerable to this attack. Note that overfitting is not the only reason, but a well-generalized model can prevent membership inference attacks. Tramèr *et al.* [9] introduced model extraction attacks, aiming to extract the parameters of a target model, and showed that sensitive information can be leaked for a specific machine learning model, called kernel logistic regression. Fredrikson *et al.* [7] proposed the model inversion attack, one of the most fatal attacks that can directly violate data privacy, where an attacker can infer parts of information in the training dataset by exploiting confidence values revealed along with predictions from target models. Furthermore, they showed that an attacker can recover recognizable images from a face recognition model. Although some basic countermeasures have been proposed, the effectiveness of the countermeasures is not clear.

In this paper, we focus on the model inversion attack for neural network models and analyze how differential privacy is resistant to this attack.

C. DIFFERENTIAL PRIVACY

Differential privacy [11], [12] establishes a strong standard for preserving privacy. Intuitively, differential privacy guarantees that a randomized algorithm behaves similarly on neighboring datasets. We say that two datasets D and D' are neighboring if they differ in one entry.

Definition 1 ((ϵ, δ)-Differential Privacy): A randomized algorithm A satisfies (ϵ, δ)-differential privacy, if for all $O \subseteq Range(A)$ and for any two neighboring datasets D and D', we have:

$$\Pr\left[A\left(D\right) \in O\right] \leq e^{\epsilon} \Pr\left[A\left(D'\right) \in O\right] + \delta.$$

Smaller values of ϵ will yield better privacy. Typically, ϵ is called a privacy budget or privacy cost. The basic definition of differential privacy is ϵ-differential privacy, which excludes the additive term δ. Although it is preferred that ϵ is less than one in pure differential privacy, this assumption is too strong and privacy budgets greater than one are applied in practice (e.g. [39], [40]). The notion of differential privacy means that the output value of a differentially private mechanism is not significantly changed in the record of any individual. There are several basic mechanisms that satisfy differential privacy, such as the Laplace mechanism and Gaussian mechanism. Differentially private mechanisms are defined in the context of sensitivity.

Definition 2: For any two neighboring datasets D and D', the l_2 sensitivity of function f is:

$$\Delta_2 f = \max_{D,D'} \left\| f\left(D\right) - f\left(D'\right) \right\|_2.$$

The sensitivity of a function is related to how much the output should be perturbed to preserve differential privacy. The Gaussian mechanism $\mathcal{M}_G$, which is one of the most common differentially private mechanisms, is defined by:

$$\mathcal{M}_G\left(D\right) = f\left(D\right) + \mathcal{N}(0, \sigma^2),$$

where $\mathcal{N}(0, \sigma^2)$ is the Gaussian distribution with mean 0 and standard deviation σ. If $\sigma > 2\ln\left(1.25/\delta\right) \cdot \Delta_2 f/\epsilon$ and $\epsilon < 1$, a single execution of the Gaussian mechanism fulfills (ϵ, δ)-differential privacy (the value of δ is preferred as smaller than $1/|D|$).

There are several properties of differential privacy which make it particularly useful, such as group privacy, composability, and resistance to post-processing. Group privacy can address differential privacy in surveys that include multiple members such as a family by linearly increasing with the size of the group. Composability, which is one of the most useful properties of differential privacy, means that if all of the components in a mechanism satisfy differential privacy, the mechanism can be differentially private in terms of a privacy budget. Thus, it allows designing mechanisms as a module. Resistance to post-processing ensures that the composition of any data-independent function with a differentially private mechanism also satisfies differential privacy.

Recently, the concept of concentrated differential privacy [41], [42] was introduced to relax the notion of traditional differential privacy. It enables tighter estimation and focuses on cumulative privacy loss with a large number of computations.

D. DIFFERENTIAL PRIVACY FOR DEEP NEURAL NETWORKS

In general, neural network models are optimized by a stochastic gradient descent algorithm (SGD) with a large amount of iteration. To preserve the privacy of neural networks, many studies have been conducted in the field of differential privacy. Differential privacy has mainly been applied to convex optimization problems or shallow models [12]–[18], [22]–[31]. In order to satisfy differential privacy for neural network models with a non-convex optimization problem, more sophisticated approaches have recently emerged.

Basically, as an approach to output perturbation, Zhang *et al.* [30] proposed not only an efficient differential privacy algorithm for the convex problem, but also a random round private stochastic gradient descent algorithm for the non-convex optimization problem with smooth objective. By using algorithmic stability arguments, they derived the sensitivity that determines the amount of noise and enabled output perturbation efficiently.

As another approach for optimizing the complex (non-convex) problem in the notion of differential privacy, methods that perturb the objective function have been recently emerged based on the functional mechanism [31] (the model is optimized for the distorted loss function). In this context, Phan *et al.* [20] focused on preserving the differential privacy of deep auto-encoders and demonstrated the usefulness of the differentially private model in human behavior prediction. For another deep learning model, Phan *et al.* [21] proposed differentially private convolutional deep belief networks

based on applying the functional mechanism to restricted Boltzman machines. However, this approach is difficult to apply to general neural network models.

As an approach that has already been applied, there are methods to distort the optimization procedure, where the parameter update is conducted with noisy gradients at each iteration. By enforcing each update (e.g., SGD) to ensure differential privacy, the final output model can satisfy differential privacy with the composition theorem. Obviously, the privacy cost increases in proportion to the number of iterations. Therefore, in this approach, the most important point to meet differential privacy for non-convex problem is to minimize the privacy cost at each iteration. In this respect, Abadi *et al.* [19] proposed a differentially private stochastic gradient descent algorithm and suggested a breakthrough composition theorem, called moments accountant, by combining Gaussian mechanisms with random sampling. They showed that differential privacy can be satisfied within moderate privacy costs. Recently, in this context, studies that extend traditional differential privacy to the notion of concentrated differential privacy have been conducted. Lee and Kifer [32] proposed a more improved gradient-based differentially private algorithm by carefully allocating the privacy budget in each iteration, and showed that it works with concentrated differential privacy. In [32], by introducing the noisy max algorithm that enables adaptive privacy budget allocation, the authors showed that the final output model can satisfy differential privacy with lower privacy costs. By improving the differentially private stochastic gradient descent algorithm [19], Yu *et al.* [33] extended traditional differential privacy to the notion of concentrated differential privacy and proposed a dynamic privacy budget allocation framework to improve the model accuracy.

In this paper, we practically analyze and evaluate how differential privacy, especially traditional (ϵ, δ)-differential privacy, behaves when applied as a countermeasure against the model inversion attack [7] in face recognition systems. Usually, face data have a large domain and face recognition is considered a complex problem. Based on these points, we focus on Abadi *et al.*'s basic differentially private stochastic gradient descent algorithm [19] for the purpose of satisfying the differential privacy of the neural network model with a non-convex optimization problem within moderate privacy costs.

III. DIFFERENTIAL PRIVACY AGAINST MODEL INVERSION ATTACK

In this section, we describe how to train models while satisfying differential privacy and illustrate the model inversion attack in detail. In addition, we introduce a new metrics that can quantify attack performance.

A. LEARNING WITH DIFFERENTIAL PRIVACY

To enforce differential privacy in neural network models, we leverage the differentially private stochastic gradient descent (DPSGD) [19]. Algorithm 1 presents the DPSGD algorithm [19]. First, a mini-batch is selected by random sampling with probability $q = L/N$, where L and N denote the size of mini-batch and dataset respectively. Then, the gradients of the loss function $\mathcal{L}$ with respect to the current model parameters θ_t are computed for each element in the mini-batch, and the computed gradients for each element are clipped by l_2-clipping with clipping parameter C. At this point, Gaussian noise is added to the aggregated gradient to satisfy differential privacy before averaging. Note that, the clipping parameter C is a hyperparameter, and in our experimental scenario, we set the clipping parameter to 4, as in [19] (the highest accuracy was measured when $C = 4$, and see [19] for more details on choosing the clipping parameter). Finally, model parameters are updated using the noisy gradients with the learning step η_t. This procedure is iterated T times and the overall privacy cost for the final output model θ_T is calculated. Also, we leverage moments accountant, where privacy costs are fine-tuned by considering the Gaussian mechanism with random sampling. The moments accountant keeps track of a bound on the moments of the privacy loss, which is a random variable.

Algorithm 1 Differentially Private Stochastic Gradient Descent

Input : Dataset $D = \{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$.

Parameters: Learning rate η_t, noise scale σ, batch size $\mathcal{L}$, clipping bound C.

Initialize θ_0 randomly

for $t \in \{1, \ldots, T\}$ **do**

Random sampling

sample L_t with probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clipping

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}\left(0, \sigma^2 C^2 \mathbf{I}\right)\right)$

Update

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output : θ_T and compute the overall privacy cost (ϵ, δ)

For neighboring datasets $D, D' \in \mathcal{D}^N$, a mechanism $\mathcal{M}$, auxiliary information aux, and an output $o \in \text{Range}(\mathcal{M})$, the λ-th moment of the privacy loss random variable at output o is defined as follows:

$$\alpha_{\mathcal{M}}\left(\lambda; aux, D, D'\right) \stackrel{\text{def}}{=} log\ \mathbb{E}_{o\sim\mathcal{M}(aux,d)}\left[exp\left(\lambda \cdot log\frac{Pr\left[\mathcal{M}(aux, D]\right.\right.}{Pr\left[\mathcal{M}(aux, D']\right.}\right)\right].$$

Then, we can state composability (for the composition) and tail bound (for the tail bound of (ϵ, δ)-differential privacy).

Theorem 1 [19]: Let $\alpha_{\mathcal{M}}(\lambda) \stackrel{\text{def}}{=} \max_{aux,D,D'} \alpha_{\mathcal{M}}(\lambda; aux, D, D')$.

1) [Composability] Suppose that a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ where

$\mathcal{M}_i : \prod_{j=1}^{i-1} \text{Range}\left(\mathcal{M}_j\right) \times \mathcal{D} \rightarrow \text{Range}\left(\mathcal{M}_i\right)$. Then, for any λ

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{i=1}^{k} \alpha_{\mathcal{M}_i}(\lambda).$$

2) [Tail bound] For any $\epsilon > 0$, the mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private for

$$\delta = \min_{\lambda} \exp\left(\alpha_{\mathcal{M}}(\lambda) - \lambda\epsilon\right).$$

By Theorem 1, we can compute the overall privacy cost by summing $\alpha_{\mathcal{M}}(\lambda)$ of each iteration. Thus, it is sufficient to bound $\alpha_{\mathcal{M}}(\lambda)$ at each iteration, and then we can compute the (ϵ, δ)-differential privacy guarantee by using the tail bound. In [19], the authors presented the asymptotic bound of $\alpha_{\mathcal{M}}(\lambda)$ as follows:

$$\alpha_{\mathcal{M}}(\lambda) \leq \frac{q^2\lambda(\lambda+1)}{(1-q)\sigma^2} + O(q^3\lambda^3/\sigma^3).$$

With the above asymptotic bound of $\alpha_{\mathcal{M}}(\lambda)$ with Theorem 1, the overall privacy cost of Algorithm 1 can be derived as follows:

Theorem 2 [19]: There exist constants c_1 and c_2 so that given the sampling probability $q = L/N$ and the number of steps T, for any $\epsilon < c_1 q^2 T$, Algorithm 1 is (ϵ, δ)-differentially private for any $\delta > 0$ if we choose

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\epsilon}.$$

Compared to the strong composition theorem, Theorem 2 can save a factor of $\sqrt{\log(T/\delta)}$ in the noise scale. This means that the overall privacy cost can be saved for the same noise scale. As a result, DPSGD with moments accountant enables optimizing deep learning models within a moderate privacy cost.

B. MODEL INVERSION ATTACK AGAINST NEURAL NETWORK MODEL

The model inversion attack is a well-known attack in adversarial use of machine learning. Given a target model, an attacker can infer certain sensitive features of the training data through the attack. Fig. 1 shows an overview of the model inversion attack on a neural network model. By converting the attack into an optimization problem, the attacker optimizes an initial vector to find the input that minimizes the error between the predicted confidence value and the ideal output value. This optimization can be performed simply by using a gradient descent algorithm. As a result, the attacker can recover parts of information from the training dataset. In particular, Fredrikson *et al.* [7] showed that it is possible to reconstruct training data (face data) in a neural network-based model for face recognition when the attacker can access a target model and knows target face labels. Algorithm 2 demonstrates the model inversion attack for a neural network-based model in a facial recognition system [7]. Let $f^{target}(\cdot)$ be a given target model and ℓ be a given target class. Initially, the algorithm sets the cost function $cost(\boldsymbol{X}) = 1 - f_{\ell}^{target}(\boldsymbol{X})$,

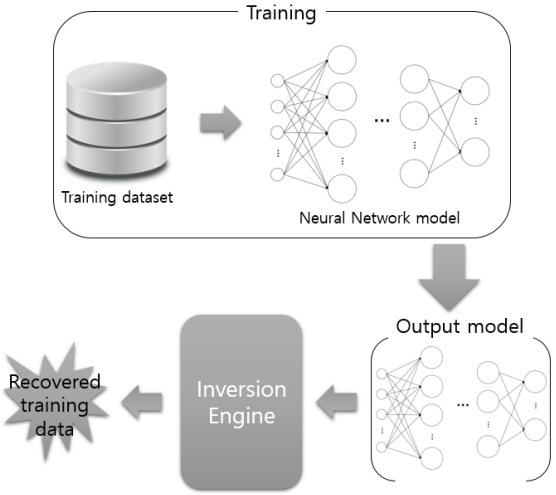

FIGURE 1. Overview of model inversion attack on neural network model.

Algorithm 2 Inversion Attack for Facial Recognition Models

Input : Target model $f^{target}(\cdot)$, target label ℓ.
Parameters: The number of iterations α, threshold β, γ, learning rate λ.

$cost(\boldsymbol{X}) \stackrel{\text{def}}{=} 1 - f_{\ell}^{target}(\boldsymbol{X})$
Initialize $\boldsymbol{X}_0 \leftarrow 0$
for $i \in \{1, \ldots, \alpha\}$ **do**
  $\boldsymbol{X}_i \leftarrow \text{Process}(\boldsymbol{X}_{i-1} - \lambda \cdot \nabla cost(\boldsymbol{X}_{i-1}))$
If $cost(\boldsymbol{X}_i) \geq \max\left(cost(\boldsymbol{X}_{i-1}), \ldots, cost\left(\boldsymbol{X}_{i-\beta}\right)\right)$ **then**
  break
If $cost(\boldsymbol{X}_i) \leq \gamma$ **then**
  break
Output : $\arg\max_{\boldsymbol{X}_i}(cost(\boldsymbol{X}_i)), \min_{\boldsymbol{X}_i}(cost(\boldsymbol{X}_i))$

FIGURE 2. An example of AT&T face data (left) of a victim used in training a face recognition model and a reconstructed face image (right) from the face recognition model using the model inversion attack.

where $f_{\ell}^{target}(\boldsymbol{X})$ indicates the output confidence value of the target model corresponding to target class ℓ with respect to input $\boldsymbol{X}$, and initializes the candidate input $\boldsymbol{X}$ that is to be optimized. Then it performs optimization through gradient descent within a given number of iterations until the cost of the candidate is not improved in β iterations or is less than the given threshold γ. Finally, the best candidate is returned. Note that Process $(\cdot)$ is a post-processing function that can execute various image manipulations. Fig. 2 shows a reconstructed example with an original face image in the AT&T face dataset [34].

By conducting model inversion attacks on differentially private models that have different privacy parameters, we explore how differential privacy is resistant to the model inversion attack.

C. ATTACK PERFORMANCE METRIC

In order to evaluate the efficacy of the model inversion attack, it is necessary to judge whether the recovered data are recognizable. In addition, it should be evaluated how many of the recovered data are recognizable and how high the attack probability is. In this respect, we present a new measuring tool and performance metrics.

1) TOOL FOR MEASURING ATTACK PERFORMANCE

To quantify the efficacy of the attack, a measuring tool that evaluates whether the reconstructed data are recognizable is essential. In [7], evaluating the recognizability of the reconstructed data (for non-private models) was performed using Amazon's Mechanical Turk, where they asked workers to match the reconstructed data from five face images or to respond that the reconstructed data does not correspond to any images. Note that these five candidate face images were extracted from the test dataset. With the survey results, the authors showed that the reconstructed data are of sufficient quality to recognize individuals. However, it is not practical to use Mechanical Turk surveys to evaluate the relationship between the model inversion attack and all differentially private models with different privacy parameters. Indeed, this survey-based evaluation requires a lot of time and resources. In other words, it has to wait for responses from a large number of workers, and pay for all responses (in [7], the author paid $ 0.05 for each response). Furthermore, it can be subjective depending on the worker. Therefore, it is not suitable to use Mechanical Turk surveys as a measuring tool for attack performance, and a more efficient and objective evaluation method is needed.

Instead of this survey-based evaluation, we leverage a deep learning model to evaluate the effectiveness of the attack. Quantifying the efficacy of the model inversion attack can be converted into a classification problem. It can be regarded as quantifying how many data is recognizable among the reconstructed data i.e., how many data have enough quality to be recognized among the reconstructed data.

Let $f^{eval}(\cdot)$ be a well-trained evaluation model that has the same classification problem as the target model $f^{target}(\cdot)$. Then, a reconstructed image extracted from the model inversion attack corresponding to a given target label is evaluated over the evaluation model $f^{eval}(\cdot)$. That is, the reconstructed image becomes the input of the evaluation model $f^{eval}(\cdot)$ in the inference phase, and then $f^{eval}(\cdot)$ returns the label (or confidence vector) as it is most likely. If the retuned label (or argmax of confidence vector) is equal to the given target label, the reconstructed image is considered to be recognizable. We quantify the efficacy of the attack over all of the reconstructed data with the above evaluation process.

From the perspective of training the evaluation model, it is assumed that the training dataset for the evaluation model and the training dataset that was used to train the target model are independent. In the original model inversion attack [7], it is assumed that an adversary cannot access the training dataset of the target model, and the evaluations (Mechanical Turk surveys) were conducted by comparing the reconstructed data with candidate face images sampled from the test dataset that is not used to train the target model. In this respect, training the evaluation model should have the same conditions as the Mechanical Turk workers. Therefore, we train the evaluation model over the test dataset that was only used to check performances of the target model.

Fig. 3 presents the overall evaluation framework. First, we separate the entire dataset into training and test datasets. Next, we generate a differentially private model (DP-model) with Algorithm 1 and moments accountant, given privacy parameters and the training dataset. Note that this private model is the target model $f^{target}(\cdot)$ of the model inversion attack and the test dataset can be used to check the performance of the target model. In parallel, we learn a deep learning model that serves as the evaluation model $f^{eval}(\cdot)$ for attack performance over the test dataset. Then we conduct the model inversion attack for the target model and all classes of the dataset (e.g., all individual names). As a result, we can obtain reconstructed face data according to the number of classes. Finally, we evaluate whether these reconstructed face data are recognizable using the evaluation model $f^{eval}(\cdot)$, i.e., the reconstructed images become inputs of the well-trained evaluation model in the inference step.

2) PERFORMANCE METRICS OF ATTACK

From the perspective of investigating the relationship between the efficacy of the model inversion attack and differential privacy, we use two metrics – success rate and impact of the attack. The success rate indicates the percentage of models successfully attacked by the model inversion attack. – We generated 100 models for each pair of parameters ϵ and σ (We created and evaluated a number of models because the attack may or may not work for a given model) and judged that the privacy of a given model was violated if at least one of the reconstructed data was classified correctly by the evaluation model. The attack impact indicates how many data are correctly classified by the evaluation model among the reconstructed data extracted from a target model, i.e., the attack impact means the accuracy of the evaluation model on the reconstructed dataset extracted from a given target model (we displayed the highest impact among the 100 models).

For a given privacy parameter, the evaluation procedure (as shown in Fig. 3) is performed for 100 models, and the success rate and highest impact are calculated.

IV. EXPERIMENTS

In [7], the authors evaluated a number of typical neural network models, including a softmax regression, a multilayer perceptron, and an auto-encoder. In our experiment,

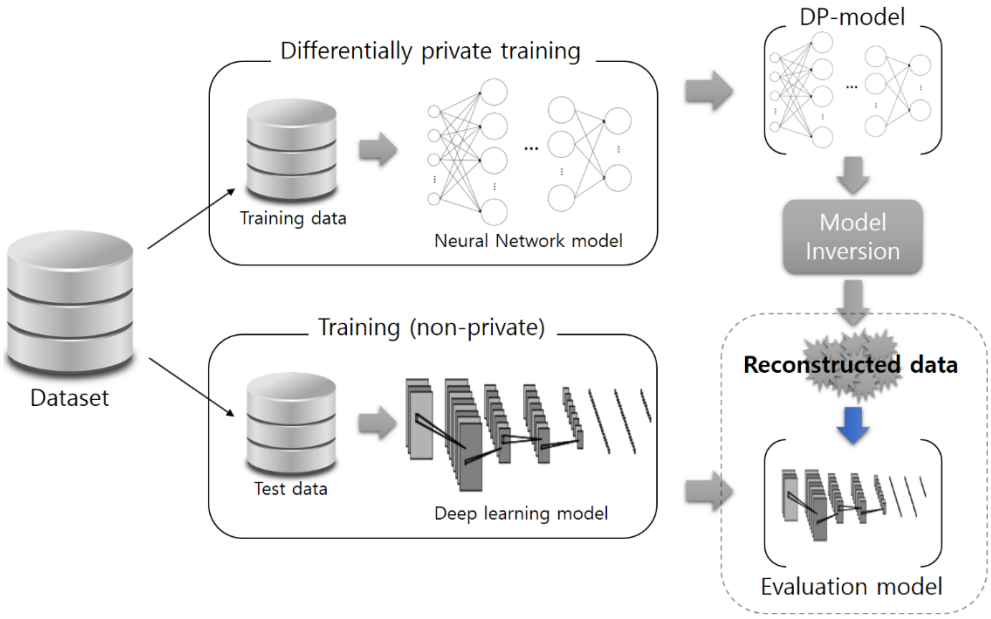

FIGURE 3. Overall framework of the evaluation procedure.

we considered the softmax regression that was most vulnerable to the model inversion attack in a non-private scenario. In general, softmax regression is used as the output layer in neural network-based architectures, and is regarded as a neural network model with no hidden layer.

In order to evaluate the resistance of differential privacy to the model inversion attack, we generated a number of target models for a given privacy parameter, performed a model inversion attack on each model, and evaluated the recognizability of the reconstructed data using an evaluation model. To build the evaluation model, we used a simple convolutional neural network (CNN) architecture with one convolution layer followed by two fully connected layers, as shown in Fig. 3 (in the center bottom box). The convolution layer contains 30 filters with a kernel size of 5×5, stride of 1, and no padding. The output of the convolution layer is fed into a 2×2 max pooling layer, and then its output is passed through a rectified linear unit (ReLU). The output of the ReLU unit is fed into two fully connected layers (each has 100 units). Finally, the output of the fully connected layer is fed into a softMax layer with the number of classes in training dataset.

We basically experimented with the AT&T Laboratories Cambridge database of faces [34], which had been used in the original model inversion attack [7] in a non-private scenario. The AT&T face dataset consists of 10 grayscale images of 40 individuals with various lighting conditions, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses) for a total of 400 images. We divided the images of each individual into a training set and a test set at a ratio of 7: 3 i.e., seven training data and three test data. Additionally, to investigate the influence on the size of the dataset, we used the VGGFace2 dataset [35], which contains a total of 3.3M images for 9.1K individuals with more varied conditions than the AT&T face dataset.

A. EXPERIMENT ON AT&T FACE DATASET

We first demonstrate the resistance of differential privacy to model conduction attacks on the AT&T face dataset.

1) TRAINING DIFFERENTIALLY PRIVATE MODELS

The size of each image of the AT&T face dataset is 9×112 pixels, with 256 grey levels per pixel. Therefore, the structure of the target model consists of 10304 input units and 40 output units (the number of individuals). We learned the model with algorithm 1, and set sampling probability $q = 0.1$ and privacy parameters to $\epsilon = \{2, 4, 6, 8\}$, $\delta = 10^{-3}$. Also, we derived the number of iterations for the fixed noise scales $\sigma = \{2, 4, 6\}$ by Theorem 2 with the privacy parameters. In a non-private scenario, the model achieved 99% accuracy in the training dataset and 93.5% accuracy in the test dataset. Table 1 presents the performance of private models with different privacy budgets and noise scales. Naturally, the greater the budget, the better the performance. The best test performances were measured at 32%, 71%, 86%, and 91% when $\epsilon = 2$, 4, 6, and 8, respectively.

2) MODEL INVERSION ATTACK ON PRIVATE MODELS

For each differentially private model, we conducted the model inversion attack and obtained 40 reconstructed face data according to the number of classes (individuals). Fig. 4 presents reconstructed face images in each private model. Compared with the reconstructed image extracted from the non-private model, it can be observed that the

TABLE 1. Training and test accuracies of differentially private models on AT&T face dataset for different privacy parameters.

			σ		
			2	4	6
ϵ	2	train	0.32	0.35	0.32
		test	0.30	0.32	0.30
	4	train	0.55	0.75	0.77
		test	0.52	0.70	0.71
	6	train	0.87	0.89	0.90
		test	0.81	0.85	0.86
	8	train	0.97	0.97	0.98
		test	0.91	0.90	0.90
Non-private				train	0.99
				test	0.93

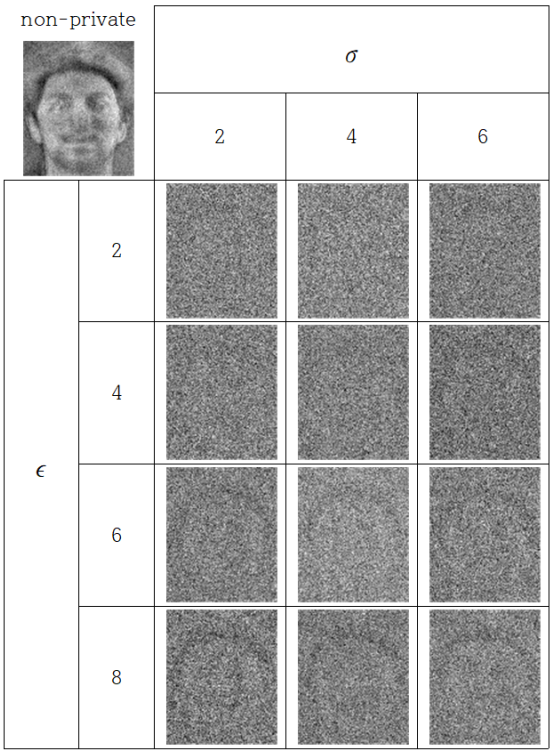

FIGURE 4. Reconstructed data extracted from differentially private models on AT&T face dataset for different privacy parameters. The top left image is a reconstructed image extracted from a non-private model.

reconstructed data extracted from the differentially private model are more blurred. At first glance, it seems enough to preserve an individual's privacy. However, as ϵ grows, it can be observed that information such as the outline of the face is revealed, and this information can be used to infer the individual.

3) ATTACK PERFORMANCE

We evaluated the recognizability of the reconstructed data using an evaluation model that was learned over the test dataset with a size of 120 (three images per class), and achieved about 85% impact on the reconstructed dataset extracted from the non-private model (the success rate is 100% in a non-private scenario). That is, all reconstructed data except six images were correctly recognized by the evaluation model. Note that the average performance (impact) with Mechanical Turk was measured to be about 80% in [7]. By using this evaluation model, we evaluated the recognizability of the reconstructed data extracted from differentially private models. Table 2 shows the results. When $\epsilon = 2, 4$, it seems to be resistant to attack, but this is due to the low accuracy of the target (differentially private) model. When $\epsilon = 6$ (the accuracy of the differentially private model is above 80%), the maximum success rate was measured at 11% and at most two reconstructed data out of 40 images were correctly recognized by the evaluation model. When $\epsilon = 8$, the success rates were measured relatively high, but the impacts were measured at 0.075 at most.

TABLE 2. Success rate and impact of the model inversion attack against differentially private models on AT&T face dataset.

			σ		
			2	4	6
ϵ	2	Success rate	0.0	0.0	0.0
		Impact	0/40	0/40	0/40
	4	Success rate	0.09	0.07	0.04
		Impact	1/40	1/40	1/40
	6	Success rate	0.11	0.09	0.07
		Impact	2/40	2/40	2/40
	8	Success rate	0.46	0.38	0.31
		Impact	3/40	2/40	2/40
Non-private		Success rate			1.0
		Impact			34/40

B. EXPERIMENT ON VGGFACE2 DATASET

The AT&T face dataset is insufficient to evaluate the resistance of differential privacy to the model inversion attack because the data set size and diversity are relatively small. Therefore, we analyzed the relationship between differential privacy and model inversion attack over an extended dataset – VGGFace2. Compared to the images of the AT&T face dataset, VGGFace2 has a larger domain. That is, the images of VGGFace2 are more diverse in facial expressions, angles, resolutions, etc. Instead of using all data, we randomly selected 40 classes from 9.1K classes, chose 100 images in each selected class (a total of 4000 images), and extracted only the face from each image by using an alignment technique.

1) TRAINING DIFFERENTIALLY PRIVATE MODELS

We set the alignment to 100×100 pixels with 256 grey levels per pixel. Therefore, the structure of the target model consisted of 10000 input units and 40 output units. Models were trained to ensure differential privacy in the same way as above and we set sampling probability $q = 0.03$,

TABLE 3. Training and test accuracies of differentially private models on VGGFace2 dataset for different privacy parameters.

			σ		
			2	4	6
ϵ	2	train	0.56	0.71	0.77
		test	0.50	0.68	0.70
	4	train	0.79	0.87	0.87
		test	0.70	0.77	0.77
	6	train	0.88	0.92	0.96
		test	0.76	0.78	0.79
	8	train	0.95	0.95	0.96
		test	0.80	0.80	0.81
Non-private				train	0.87
				test	0.825

FIGURE 5. An example of VGGFace2 data (left) of a victim used in training a face recognition model and a reconstructed face image (right) from the face recognition model using the model inversion attack.

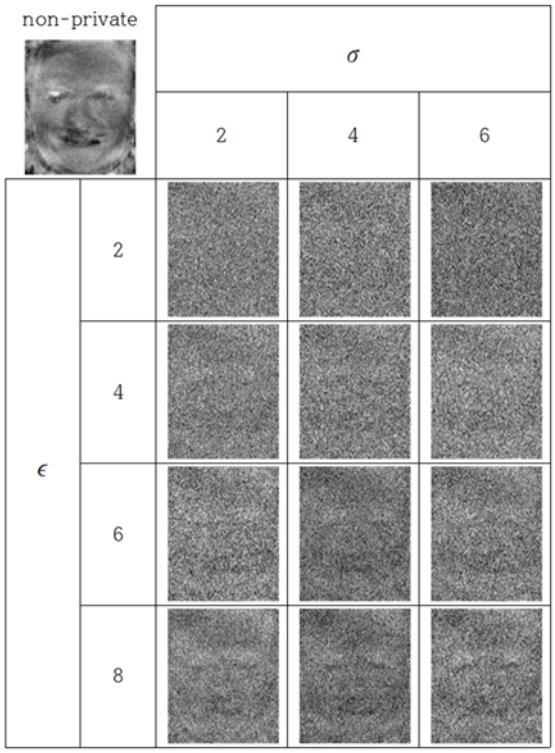

FIGURE 6. Reconstructed data extracted from differentially private models on VGGFace2 for different privacy parameters. The top left image is a reconstructed image extracted from a non-private model.

$\epsilon = \{2, 4, 6, 8\}$, $\delta = 10^{-4}$, and $\sigma = \{2, 4, 6\}$. In a non-private scenario, the model achieved 87% accuracy in the training dataset and 82.5% accuracy in the test dataset. Table 3 shows the performance of private models with different privacy parameters. The best test performances were measured at 70%, 77%, 79%, and 81% when $\epsilon = 2$, 4, 6, and 8, respectively. When $\epsilon = 2$ and 4, the performances of the differential private models were improved in both absolute and relative terms compared to the results in the AT&T face dataset. When $\epsilon = 6$ and 8, the performances are absolutely poor, but we can know that it has better performances in relative comparison to the non-private model, i.e., the distances between the non-private model and the differentially private model are small. These results are expected to come from the size of the dataset. In this situation, we investigated how differential privacy is resistant to model inversion attacks.

2) MODEL INVERSION ATTACK ON PRIVATE MODELS

Similar to the case of the AT&T face dataset, we extracted face images from non-private models and each differentially private model. Fig. 5 presents an example of reconstructed data and original image pair in a non-private model, and Fig. 6 shows reconstructed images in each differentially private model. The reconstructed data extracted from the differentially private models are more blurred than the reconstructed data extracted from the non-private model, and it can be seen that the outline of the face gradually becomes visible as ϵ grows.

3) ATTACK PERFORMANCE

In order to evaluate the recognizability of the reconstructed data, we learned the evaluation model in the test dataset with a size of 1200 (30 images per class), and achieved about 82.5% impact on the reconstructed dataset extracted from the non-private model, i.e., all reconstructed images were correctly recognized by the evaluation model except seven images (the success rate is 100% in a non-private scenario). By using this evaluation model, we evaluated the recognizability of the reconstructed images extracted from the private models, and measured the attack accuracy and impact in the same manner as the experiments with the AT&T face dataset. Table 4 shows the results. When $\epsilon = 2$, 4, and 6, attack success rates were measured at less than 3%, 10%, and 15%, and impact was measured at less than 2%, 3%, and 5%, respectively. When $\epsilon = 8$, more than half of the private models were attacked for small sigma, and the attack impact was measured at 37.5%.

C. EVALUATION

In both experiments on the AT&T face dataset and VGGFace2, we observed that, compared to a non-private model, differentially private models dramatically reduced the probability of model inversion attack. Note that in the non-private scenario, the attack impact on the target model is 85% and 82.5% for the AT&T face dataset and VGGFace2, respectively. Also, we are confident that the attack success rate is 100% for non-private models. In the cases where ϵ is 6 or less, we can show that differentially private models

TABLE 4. Success rate and impact of the model inversion attack against differentially private models on VGGFace2 dataset.

			σ		
			2	4	6
ϵ	2	Success rate	0.00	0.01	0.03
		Impact	0/40	1/40	2/40
	4	Success rate	0.10	0.10	0.08
		Impact	3/40	3/40	2/40
	6	Success rate	0.15	0.12	0.12
		Impact	5/40	4/40	4/40
	8	Success rate	0.52	0.46	0.41
		Impact	15/40	11/40	12/40
Non-private		Success rate	1.0		
		Impact	33/40		

can prevent model inversion attacks with high probability. When $\epsilon = 8$, almost half of the models were attacked with a maximum 37.5% attack impact. The attack success rate is reduced to almost half, and the attack impact is reduced to a maximum of 0.05% and 27.5% (from 85% and 82.5%).

V. RELATED WORK

Various countermeasures against privacy breach attacks on the machine learning model have been proposed. Rahman *et al.* [43] studied the relationship between differential privacy and the membership inference attack [8]. By using the DPSGD [19], they empirically investigated the appropriate value of ϵ to find the best trade-off between utility and privacy while preventing membership inference attacks. However, it is unclear whether the results come from differential privacy or from being generalized by it (generalization can mitigate a membership inference attack). It is necessary to compare the scenario where target models satisfy differential privacy and the scenario in which only generalization is considered.

Fredrikson *et al.* [6] proposed the model inversion attack where a patient's privacy can be violated by the machine learning models used for medical treatments in pharmacogenetics. They experimented with a linear regression model and showed that differential privacy can prevent attacks only when the privacy budget is very small. In particular, they focused on generating a differentially private histogram and differentially private linear regression algorithms. In order to ensure differential privacy while preserving utility in the pharmacogenetics analysis, Wang *et al.* [44] proposed an approach that separates sensitive and non-sensitive attributes by leveraging the functional mechanism for regression models.

In a privacy preservation study of the recent model inversion attack [7], Zhang [45] proposed obfuscation methods where noise is added to the training data before training the model and showed that the recovered data are more blurred compared to the reconstructed data from a non-private model. However, there is a lack of consideration as to how this approach can withstand the model inversion attack while retaining model utility.

TABLE 5. Comparison of recognition rates of reconstructed data for different evaluation metrics in the non-private scenario on the AT&T face dataset and VGGface2 dataset.

	AT&T face data	VGGFace2
Survey-based approach [7]	0.80	-
SSIM [46]	0.775	0.6
Our method	0.85	0.825

From the perspective of the model inversion attack, one can consider other approaches to assess the recognizability of reconstructed images. Traditionally, in the field of image processing, well-known metrics such as the structural similarity index (SSIM) [46] have been used to assess the quality of images. These metrics measure the similarity or difference between a given original image x and the transformed data y from the original data based on the distance, and one can consider these tools as a way of resolving the problem addressed in this paper (label inferencing of reconstructed data extracted from face recognition model). These distance-based tools determine visibility, and utilizing these tools for classifying the correct class of transformed data y may not be appropriate. As shown in Table 5, when SSIM was used for the problem of classifying reconstructed data extracted through the model inversion attack, the accuracy was measured lower than the survey-based approach conducted in [7], and the SSIM metric showed a significant decrease in performance with the VGGFace2 dataset. Hence, it may not be appropriate to use these tools. Note that all the evaluations presented in Table 5 were conducted under the same adversarial assumption that an adversary cannot access the training dataset of the target model.

VI. CONCLUSION AND FUTURE WORK

In this paper, we applied state-of-the art differentially private learning to neural network model as a way to withstand the recent model inversion attack [7] and analyzed the resistance of differential privacy depending on different privacy parameters by introducing an attack-based evaluation method using a deep learning model. By conducting model inversion attacks on differentially private neural network models, we showed that differential privacy can dramatically reduce the probability of attack compared to non-private models. In our experiments, we showed the trade-off of model utility and privacy for various privacy parameters and investigated the influence on the size of the dataset. In the case of relatively low privacy budgets, the model shows strong resistance to the model inversion attack, but accuracy is

sacrificed to preserve strong privacy. In the case of relatively large privacy budgets, attack success rates and impacts are measured as somewhat higher compared to the case of low privacy budgets, but we confirmed that attack success rates and impacts are significantly lower than in the non-private scenarios. We believe that our attack-based evaluation can be used as an important indicator to measure the degree of privacy preservation when creating private neural network models. Note that although we leveraged Abadi et al.'s differentially private SGD [19] algorithm and the moments accountant in this paper, a tighter bound can be calculated with more optimal building blocks (e.g., analytic Gaussian mechanism [47] and analytic moments accountant [48]). As a future direction, it would be interesting to compare various differentially private mechanisms for neural networks by applying our evaluation method.

While this paper focused on models with the non-convex objective, many problems are solved with basic machine learning models that have convex objectives. Differentially private mechanisms have been proposed for these baseline machine learning models, and there are efficient approaches tailored to the type of machine learning model. We will investigate the application of our attack-based evaluation approach to various differentially private machine learning models.

Additionally, although we focused on traditional differential privacy, it would be interesting to analyze the attack probability of the model inversion attack through our evaluation framework in the scenario where the notion of differential privacy is extended to a relaxed notion. Concentrated differentially private algorithms that can preserve the utility and privacy of deep models with a relatively low privacy budget have been recently proposed with the notion of concentrated differential privacy [41], [42] (e.g., [32], [33]). These mechanisms efficiently allocate privacy budgets and estimate a tighter bound under concentrated differential privacy. As a future extension, it would be interesting to analyze the relationship between the degree of privacy guarantee and utility from the perspective of concentrated differential privacy through our attack-based private model evaluation framework.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.
[2] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[4] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2773–2781.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[6] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 17–32.
[7] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1322–1333.
[8] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 3–18.
[9] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 601–618.
[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.*, 2006, pp. 265–284.
[11] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.
[12] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. STOC*, vol. 9, 2009, pp. 371–380.
[13] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 627–636.
[14] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 289–296.
[15] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.
[16] 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.
[17] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci.*, Oct. 2014, pp. 464–473.
[18] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proc. ACM Int. Conf. Manag. Data*, May 2017, pp. 1307–1322.
[19] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.
[20] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 1309–1316.
[21] N. H. Phan, X. Wu, and D. Dou, "Preserving differential privacy in convolutional deep belief networks," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1681–1704, 2017.
[22] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *Proc. Conf. Learn. Theory*, Jun. 2012, pp. 1–24.
[23] D. Kifer, A. Smith, and A. Thakurta, "Private convex empirical risk minimization and high-dimensional regression," in *Proc. Conf. Learn. Theory*, Jun. 2012, pp. 1–25.
[24] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for SVM learning," *J. Privacy Confidentiality*, vol. 4, pp. 65–100, Jan. 2012.
[25] K. Talwar, A. G. Thakurta, and L. Zhang, "Nearly-optimal private LASSO," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3025–3033.
[26] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2719–2728.
[27] Y.-X. Wang, S. E. Fienberg, and A. J. Smola, "Privacy for free: Posterior sampling and stochastic gradient Monte Carlo," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2493–2502.
[28] O. Williams and F. McSherry, "Probabilistic inference and differential privacy," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 2451–2459.
[29] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett, "PrivGene: Differentially private model fitting using genetic algorithms," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 665–676.
[30] J. Zhang, K. Zheng, W. Mou, and L. Wang, "Efficient private ERM for smooth objectives," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3922–3928.
[31] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012.
[32] 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*, Jul. 2018, pp. 1656–1665.

[33] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. IEEE Symp. Secur. Privacy*, May 2019, pp. 1–18.
[34] AT&T Laboratories Cambridge. *The ORL Database of Faces*. [Online]. Available: http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.tar.Z
[35] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2018, pp. 67–74.
[36] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genet.*, vol. 4, no. 8, 2008, Art. no. e1000167.
[37] G. Ateniese, G. Felici, L. V. Mancini, A. Spognardi, A. Villani, and D. Vitali, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," 2013, *arXiv:1306.4447*. [Online]. Available: https://arxiv.org/abs/1306.4447
[38] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, "'You might also like:' Privacy risks of collaborative filtering," in *Proc. IEEE Symp. Secur. Privacy*, May 2011, pp. 231–246.
[39] Differential Privacy Team, "Learning with privacy at scale," *Apple Mach. Learn. J.*, vol. 1, no. 8, pp. 1–25, Dec. 2017.
[40] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1054–1067.
[41] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*. [Online]. Available: https://arxiv.org/abs/1603.01887
[42] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptogr. Conf.*, Nov. 2016, pp. 635–658.
[43] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model," *Trans. Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.
[44] Y. Wang, C. Si, and X. Wu, "Regression model fitting under differential privacy and model inversion attack," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 1003–1009.
[45] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving machine learning through data obfuscation," 2018, *arXiv:1807.01860*. [Online]. Available: https://arxiv.org/abs/1807.01860
[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[47] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," 2018, *arXiv:1805.06530*. [Online]. Available: https://arxiv.org/abs/1805.06530
[48] Y.-X. Wang, B. Balle, and S. Kasiviswanathan, "Subsampled Rényi differential privacy and analytical moments accountant," 2019, *arXiv:1808.00087*. [Online]. Available: https://arxiv.org/abs/1808.00087

CHEOLHEE PARK received the B.S. and M.S. degrees from the Department of Applied Mathematics and Mathematics, Kongju National University, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Mathematics, Kongju National University. His research interests include cryptography, data privacy, differential privacy, machine learning, and deep learning.

DOWON HONG received the B.S., M.S., and Ph.D. degrees in mathematics from Korea University, Seoul, South Korea, in 1994, 1996, and 2000, respectively. He was a Principal Member of Engineering Staff with the Electronics and Telecommunications Research Institute, South Korea, from 2000 to 2012. In 2012, he joined the Department of Applied Mathematics, Kongju National University, South Korea, where he has been a Full Professor, since 2015. His research interests include cryptography, data privacy, and differential privacy.

CHANGHO SEO received the B.S., M.S., and Ph.D. degrees in mathematics from Korea University, Seoul, South Korea, in 1990, 1992, and 1996, respectively. He is currently a Full Professor with the Department of Applied Mathematics, Kongju National University, South Korea. His research interests include cryptography, information security, data privacy, and system security.