Review Article

Maoguo Gong, Yu Xie, Ke Pan, and Kaiyuan Feng
School of Electronic Engineering, Xidian University, Xi'an, CHINA

A. K. Qin
Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, AUSTRALIA

A Survey on Differentially Private Machine Learning

Abstract

Recent years have witnessed remarkable successes of machine learning in various applications. However, machine learning models suffer from a potential risk of leaking private information contained in training data, which have attracted increasing research attention. As one of the mainstream privacy-preserving techniques, differential privacy provides a promising way to prevent the leaking of individual-level privacy in training data while preserving the quality of training data for model building. This work provides a comprehensive survey on the existing works that incorporate differential privacy with machine learning, so-called differentially private machine learning and categorizes them into two broad categories as per different differential privacy mechanisms: the Laplace/Gaussian/exponential mechanism and the output/objective perturbation mechanism. In the former, a calibrated amount of noise is added to the non-private model and in the latter, the output or the objective function is perturbed by random noise. Particularly, the survey covers the techniques of differentially private deep learning to alleviate the recent concerns about the privacy of big data contributors. In addition, the research challenges in terms of model utility, privacy level and applications are discussed. To tackle these challenges, several potential future research directions for differentially private machine learning are pointed out.

LICENSED BY INGRAM PUBLISHING

I. Introduction

Machine learning aims to simulate the behaviors of human beings and give computers the ability to learn new knowledge or skills from data without being explicitly programmed. In the past decades, it has led to remarkable breakthroughs in both academia and industry including a variety of exciting real-world domains of images, videos, text, speech, complex networks, robots, healthcare and many more. Deep learning [1] based on artificial neural networks has rapidly become a popular branch of machine learning techniques since 2012 by virtue of the unprecedented performance. Machine learning methods, including deep learning methods, require more or less representative datasets for learning desirable models. However, these datasets may contain sensitive individual information. For the text typed on a mobile device, the individual information may include schedules, profiles, usernames, passwords, text dialogues, search queries and medical histories, etc. These data are usually privacy-sensitive even with legal and ethical constraints. With the era of big data coming, a dizzying array of semantic-rich data of individuals are being collected for analyzing, understanding and eventually entailing tremendous commercial value. In some areas such as targeted advertisements and personalized recommendations, the datasets used for machine learning tasks

Digital Object Identifier 10.1109/MCI.2020.2976185
Date of current version: 10 April 2020

Corresponding Author: Maoguo Gong
(Email: gong@ieee.org).

1556-603X/20©2020IEEE

are numerous. In some domains, especially medicine and finance, each institute only has access to limited amount of data, and large datasets are often crowdsourced. Even though these datasets for machine learning tasks enable faster commercial or scientific progress, the critical and sensible demand for preserving individual privacy from invasion continues to rise in the crowd, companies and the government.

Ideally, the sensitive individual information should not be leaked in the process of training machine learning models. In other words, we allow the parameters of machine learning models to learn general patterns (people who smoke are more likely to suffer from lung cancer), rather than facts about specific training samples (he had lung cancer). Unfortunately, shallow models like support vector machine and logistic regression are capable of memorizing secret information of the training dataset [2]. Deep models like convolutional neural networks are able to exactly memorize arbitrary labels of the training data [3]. Recent attacks against machine learning models as in [2], [4]–[8] emphasize the implicit risks and catalyze an urgent demand for privacy preserving. For examples, Shokri *et al.* [4] designed a membership inference attack that can estimate whether the training dataset contains a specific data record via the black-box access to the model. Fredrikson *et al.* [5] presented a model inversion attack that can reveal individual faces given the API of the face recognition system and the name of the user to be identified. In [6], the decision probability of the classification model can be used for the model extraction attack, which implicitly steals the sensitive training data. Attackers abuse the pharmacogenetic model to inversely infer the patients' genetic markers [7]. Adversaries can maliciously acquire unexpected but useful information from the machine learning classifiers [8].

To alleviate the possible privacy threats to data owners, an attractive and viable method is to involve the privacy-preserving techniques into machine learning approaches. In early works, it is prevalent to anonymize the data before analyzing data including k-anonymity [9], l-diversity [10], t-closeness [11], which remove private details or replace them with random values. Nevertheless, they are not always sufficient and only provide privacy guarantee to a certain extent, especially when adversaries own auxiliary individual information in the sensitive dataset. Besides, anonymizing is not applicable to high-dimensional or diverse input datasets due to its strong theoretical and empirical limitations [12], [13]. As a solid privacy model, differential privacy [14] has recently been considered as a promising strategy for privacy preserving in machine learning. There are roughly three major reasons: (1) Differential privacy can provide a provable privacy guarantee for individuals, which benefits from the most solid theoretical basis compared with other privacy-preserving models [9]–[11], [15], [16]. (2) Differential privacy achieves privacy preserving in machine learning by adding a calibrated amount of noise to the model or output results according to the concrete mechanisms instead of simply anonymizing the individual data. (3) Differential privacy can make a graceful compromise between privacy and utility by adjusting a privacy budget index, in which the smaller the value of the privacy budget, the stronger privacy guarantee it provides. For data owners, differentially private machine learning further ensures that the adversaries are incapable to infer any information about a single record with high confidence from the released machine learning models or output results, even if an adversary knows all the remaining records in this dataset. A graphical illustration of incorporating differential privacy into machine learning for privacy preserving is shown in Figure 1.

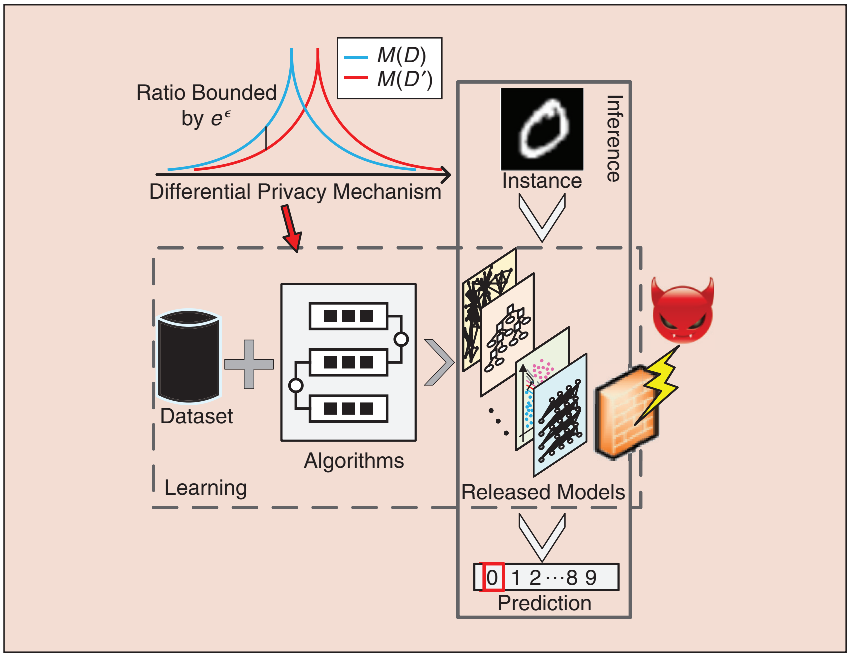

FIGURE 1 A graphical illustration of incorporating differential privacy into machine learning for privacy preserving.

In the past decade, we have witnessed the rapid advances of new methods about differentially private machine learning. This is entirely due to the remarkable capability of differential privacy in providing effective and efficient approaches for solving the problem of privacy preserving, by utilizing the basic mechanisms such as Laplace mechanism [14], exponential mechanism [17], and functional perturbation mechanism [18]. These differential privacy mechanisms can combine the strength of differential privacy to satisfy the requirement of privacy preserving for non-private prototypes of a wide range of machine learning techniques. There are only a few existing reviews on the topic of differentially private machine learning. Dwork *et al.* [19] briefly summarized a limited number of the most basic problems on differentially private machine learning, including the sample complexity, online learning and empirical risk minimization. Ji *et al.* [20] focused on generalizing differential privacy mechanisms to traditional machine learning models, while missing many newly developed approaches that update state-of-the-art benchmarks. Furthermore, both of them do not cover the advances of differential privacy in deep learning. Inspired by this, the purpose of this survey is to present a systematic and comprehensive overview of the extensive researches about differentially private machine learning. More specifically, this survey investigates on applications of different differential privacy mechanisms from the perspectives of both traditional machine learning and recently booming deep learning. Figure 2 roughly provides a summary of differentially private machine learning research. In particular, this survey has three major contributions as follows.

1) We propose a taxonomy of existing techniques about differentially private machine learning. To the best of our knowledge, we are the first to investigate the development of differentially private deep learning.
2) We provide a detailed and thorough study of the state-of-the-art methods. As a result, this survey brings new perspectives to better understand the existing works and improve the privacy level of machine learning models.
3) To facilitate the timely and potential research of this area for researchers from privacy preserving, especially machine learning, we summarize the limitations and challenges of current research works, and suggest several promising future research directions.

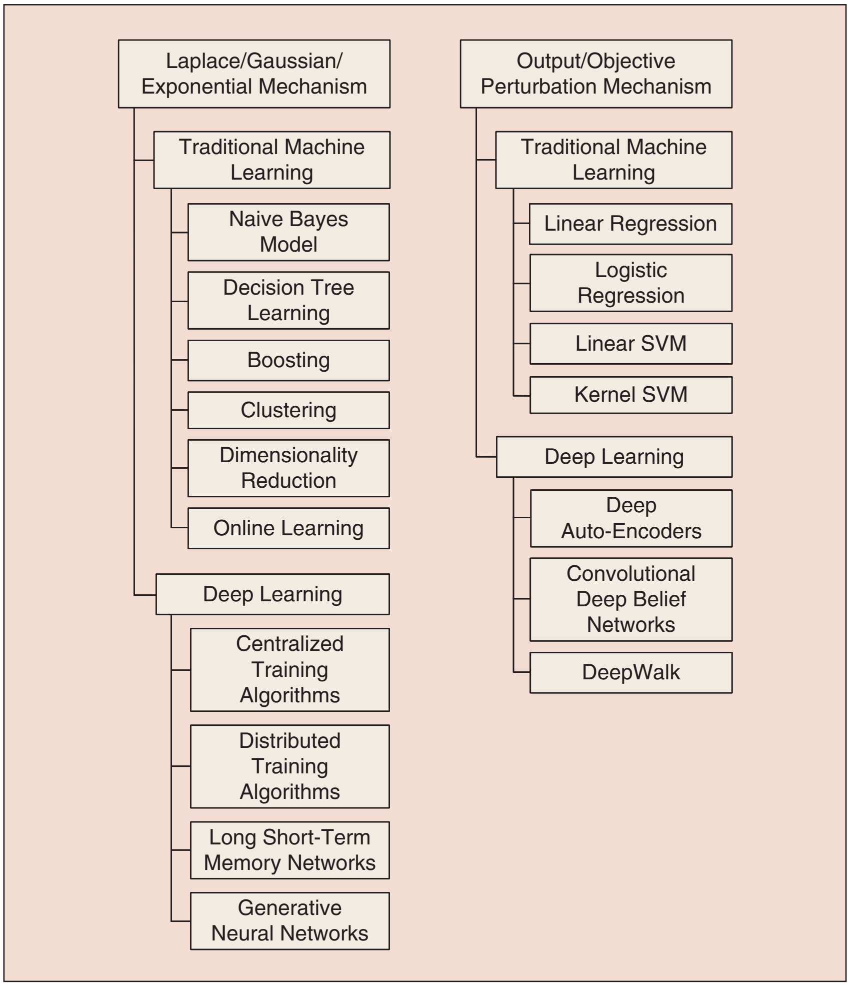
Laplace/Gaussian/ Exponential Mechanism
Traditional Machine Learning
Naive Bayes Model
Decision Tree Learning
Boosting
Clustering
Dimensionality Reduction
Online Learning
Deep Learning
Centralized Training Algorithms
Distributed Training Algorithms
Long Short-Term Memory Networks
Generative Neural Networks
Output/Objective Perturbation Mechanism
Traditional Machine Learning
Linear Regression
Logistic Regression
Linear SVM
Kernel SVM
Deep Learning
Deep Auto-Encoders
Convolutional Deep Belief Networks
DeepWalk

FIGURE 2 The proposed taxonomy to summarize the methods about differentially private machine learning.

The rest of this paper is organized as follows. Section II introduces the related backgrounds. In Section III, we give a theoretical description of the application of differential privacy in traditional machine learning in detail. Section IV carefully covers recent models of differentially private deep learning. In Section V, we discuss the existing challenges and point out promising future directions. Section VI summarizes this review. Due to the limitation of space, five summary tables are shown in the supplementary material[1].

II. Backgrounds

In this section, we will introduce the definition of differential privacy, the principle differential privacy mechanisms and then present a brief overview of machine learning. The notations used in this survey are summarized in Table I.

A. Differential Privacy

Differential privacy [14] is a solid privacy-preserving model presented by Dwork *et al.* in 2006. It aims to preserve

TABLE 1 **The common notations used in this survey.**

NOTATIONS	EXPLANATION
D	DATASET
$\mathcal{X}$	SAMPLE SPACE
$\mathcal{Y}$	OUTPUT SPACE
$\mathcal{M}$	ALGORITHM
n	SIZE OF THE DATASET
d	DIMENSION OF THE SAMPLE SPACE
$\mathcal{L}$	LOSS FUNCTION
ϵ	PRIVACY BUDGET
δ	POSSIBILITY OF VIOLATING ϵ-DIFFERENTIAL PRIVACY
$S(\cdot)$	SENSITIVITY
$Lap(\cdot)$	LAPLACE DISTRIBUTION
$\mathcal{N}(0,\sigma^2)$	GAUSSIAN DISTRIBUTION WITH MEAN 0 AND VARIANCE σ^2

[1] The supplementary material that includes five summary tables is available at http://see.xidian.edu.cn/faculty/mggong/publication.htm

privacy of each record in the dataset and does not depend on any background knowledge of adversaries.

Definition 1. (Differential Privacy) For two datasets D and D' differing only in one element, a randomized algorithm $\mathcal{M}$ guarantees (ϵ, δ)-differential privacy for any subset of the output S, if $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(D') \in S] + \delta. \quad (1)$$

The parameter ϵ denotes the privacy budget, which controls the privacy level of $\mathcal{M}$. To preserve privacy, the algorithm $\mathcal{M}$ randomizes the output and ensures that the probability of outputting the same results will not change significantly, when any record is deleted from the dataset. In specific, the probability ratio is bounded by $\exp(\epsilon)$. For a small ϵ, the probability distributions of output results of $\mathcal{M}$ on D and D' are extremely similar and it is difficult for attackers to distinguish the two datasets. Especially when $\epsilon = 0$, it implies the strongest privacy level. In addition, it provides a possibility to violate ϵ-differential privacy by a small probability δ [21]. When δ is equal to zero in Eq. 1, the randomized algorithm $\mathcal{M}$ guarantees ϵ-differential privacy.

To analyze the privacy budget consuming of composite algorithms, two differential privacy composition theorems are widely used.

Theorem 1. (Sequential Composition Theorem [17]) Suppose $\mathcal{M}$ is a set of privacy algorithms, $\mathcal{M}$ satisfies $(\Sigma\epsilon_i, \Sigma\delta_i)$-differential privacy if $\mathcal{M}$ is sequentially performed on an entire dataset and $\mathcal{M}_i \in \mathcal{M}$ satisfies (ϵ_i, δ_i)-differential privacy.

Theorem 2. (Parallel Composition Theorem [22]) Suppose $\mathcal{M}$ is a set of privacy algorithms, $\mathcal{M}$ satisfies $\max\{\epsilon_i\}$-differential privacy if $\mathcal{M}_i \in \mathcal{M}$ provides (ϵ_i, δ_i)-differential privacy guarantees on a disjointed subset of the entire dataset.

The sequential composition theorem is applicable to the scenario that a series of privacy algorithms are sequentially trained on a common dataset. The ultimate privacy budget is equal to the total privacy budgets. However, the parallel composition theorem is critical for the case that multiple privacy algorithms are separately trained on the respective subset of the entire dataset. The ultimate privacy budget depends on the maximum of privacy budgets.

In general, differential privacy can be achieved by adding a reasonable amount of noise into the output results of the query function. The amount of noise will affect the trade-off between privacy and utility of the dataset. Specifically, too much noise will make the dataset useless and too little noise is not enough for providing privacy guarantees. The amount of noise can be determined by the sensitivity. There are two types of sensitivity including the global sensitivity and the local sensitivity defined as follows.

Definition 2. (Global Sensitivity [14]) Given a query function $f: D \to \mathbb{R}$, the global sensitivity of f is defined as

$$\mathrm{GS}(f, D) = \max_{D, D'} \| f(D) - f(D') \|_1. \quad (2)$$

Definition 3. (Local Sensitivity [23]) Given a query function $f: D \to \mathbb{R}$, the local sensitivity of f is defined as

$$\mathrm{LS}(f, D) = \max_{D'} \| f(D) - f(D') \|_1. \quad (3)$$

$\| f(D) - f(D') \|_1$ represents the Manhattan distance between $f(D)$ and $f(D')$. The global sensitivity and the local sensitivity provide us with the magnitude that only one record can change the query result of f in the worst case. However, the former is independent of datasets and is only determined by the query function f whereas the latter takes both datasets and the function into consideration. The relationship between them is denoted as $\mathrm{GS}(f, D) = \max_{D} \mathrm{LS}(f, D)$. The global sensitivity works well when the sensitivity of the query function f is relatively small. If the global sensitivity is relatively large, a great amount of noise would be added into the query results for differential privacy, which may provide excessive guarantees of privacy. As for the local sensitivity, using it directly may lead to the information disclosure of the individual data.

Three fundamental mechanisms can be used to guarantee differential privacy: the Laplace mechanism [14], the Gaussian mechanism [19], and the exponential mechanism [19]. The Laplace mechanism and the Gaussian mechanism are widely used to achieve differential privacy for numerical results while the exponential mechanism is used for nonnumeric results.

Definition 4. (Laplace Mechanism [14]) For a query function $f: D \to \mathbb{R}$, a randomized algorithm $\mathcal{M}$ satisfies ϵ-differential privacy if

$$\mathcal{M}(D) = f(D) + Lap\left(\frac{S(f)}{\epsilon}\right), \quad (4)$$

where $S(f)$ denotes the sensitivity of f and $Lap(S(f)/\epsilon)$ represents the noise drawn from the Laplace distribution with the center of 0 and the scaling of $(S(f)/\epsilon)$.

Definition 5. (Gaussian Mechanism [19]) For a query function $f: D \to \mathbb{R}$, a randomized algorithm $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy if

$$\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2), \quad (5)$$

where $\mathcal{N}(0, \sigma^2)$ indicates that the noise variable is i.i.d. the Gaussian distribution with the standard deviation of $\sigma = S_2(f)\sqrt{(2\ln(2/\delta))}/\epsilon$. In the Gaussian mechanism, the l_2 sensitivity of f is used to achieve differential privacy, which is defined as $S_2(f) = \| f(D) - f(D') \|_2$ where $\| f(D) - f(D') \|_2$ represents the Euclidean distance between $f(D)$ and $f(D')$.

Definition 6. (Exponential Mechanism [17]) Suppose $q(D, r)$ is a utility function of the output r, Δq is the global sensitivity of the utility function $q(D, r)$ and O is the output domain of a randomized algorithm $\mathcal{M}$ on the dataset D. $\mathcal{M}$ satisfies ϵ-differential privacy if it returns $r (r \in O)$ with the probability proportional to $\exp(\epsilon q(D, r) / 2\Delta q)$.

B. Machine Learning

Machine learning is a prevalent paradigm for automatically discovering patterns in data and using the patterns to make predictions. Generally, machine learning aims to learn a deterministic function $f: \mathcal{X} \to \mathcal{Y}$ from the sample space $\mathcal{X}$ to the output space $\mathcal{Y}$. The machine learning algorithms can be broadly divided into two categories: supervised learning and unsupervised learning. Supervised learning typically refers to deducing the hidden pattern or function from labeled training data. Classic examples of supervised learning models include naive Bayes model, decision tree learning, linear regression, logistic regression and support vector machine (SVM), etc. Unsupervised learning aims to build a mathematical model from unlabeled data. Traditional unsupervised learning algorithms include clustering, dimensionality reduction and so on. Note that the goal of machine learning is to learn a generalized model that can perform well on the samples outside the training data. Even more, machine learning models with the strong generalization ability can be well suitable for the entire sample space. In other words, machine learning models should extract the useful information from the distribution of data on hand, rather than depend on specifics of any individual sample. Therefore, differential privacy is not in conflict with machine learning and has been successfully applied to machine learning for privacy preserving.

In deep learning, deep neural networks learn composite and highly abstract features through multiple layers and nonlinear processing units [24]. The representation ability of deep learning models raises exponentially with the increase of the number of network layers [25]. The existing deep learning models consist of convolutional neural networks, deep belief networks and recurrent neural networks, etc. These models have made great progress in numerous applications such as object detection, speech recognition, natural language understanding, medical diagnosis, social network analysis, automatic driving, board games, bioinformatics and so on. The training of deep learning models can be supervised or unsupervised, which depends on the specific task. As a branch of machine learning, deep learning is faced with the privacy issue as well. An ideal deep learning model should be conducive to the data analysis while preserving the privacy of sensitive data.

III. Application of Differential Privacy in Traditional Machine Learning

In this section, the application of differential privacy in traditional machine learning algorithms is categorized into two broad categories according to different noise perturbation mechanisms. The Laplace/Gaussian/exponential mechanism focuses on incorporating the Laplace mechanism or the Gaussian mechanism or the exponential mechanism into non-private learning models directly while the output/objective perturbation mechanism performs by adding noise to the output results or the objective function. The overall framework of incorporating differential privacy into traditional machine learning algorithms is shown in Figure 3.

A. Laplace/Gaussian/Exponential Mechanism

The Laplace mechanism, the Gaussian mechanism and the exponential mechanism are three classical differential privacy mechanisms. The privacy of individual data can be preserved by combining the Laplace mechanism or the Gaussian mechanism or the exponential mechanism with specific machine learning algorithms.

1) Supervised Learning

Naive Bayes model. Consider a given dataset $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})\}$ with $d+1$ attributes $X_1, X_2, \ldots, X_d, Y$, where $Y \in \mathcal{Y}$ is the output and $\mathcal{Y} = \{c_1, \ldots, c_k\}$ is a set of responses. The Naive Bayes model is known as a classification method based on the Bayes theorem $P(Y = c_k \mid X = \boldsymbol{x}) = (P(X = \boldsymbol{x} \mid Y = c_k)$ $P(Y = c_k) / \Sigma_k P(X = \boldsymbol{x} \mid Y = c_k) P(Y = c_k))$ and the conditional independence assumption $P(X = \boldsymbol{x} \mid Y = c_k) = \prod_{j=1}^{n} P(X_j = \boldsymbol{x}_j \mid Y = c_k)$. The Bayes theorem is used to find the output y with the largest posterior probability:

$$y = \operatorname*{argmax}_{c_k} P(Y = c_k) \times \prod_{j=1}^{n} P(X_j = \boldsymbol{x}_j \mid Y = c_k). \tag{6}$$

A pioneering method of differentially private Naive Bayes classification is proposed in [26], which derives the sensitivity for each attribute appropriately based on whether it is categorical or numeric. For categorical attributes, given an attribute X with r possible attribute values $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r$, the probability is $P(X = \boldsymbol{x}_k \mid Y = c_j) = (n_{kj}/n)$ where n is the number of total training examples and n_{kj} is the number of the training examples that also have $X = \boldsymbol{x}_k$. Moreover, the sensitivity can be calculated on the counts or on the likelihoods. Therefore, the sensitivity of each n_{kj} is 1 for all the values of attribute $\boldsymbol{x}_k$ and the values of class c_j. For

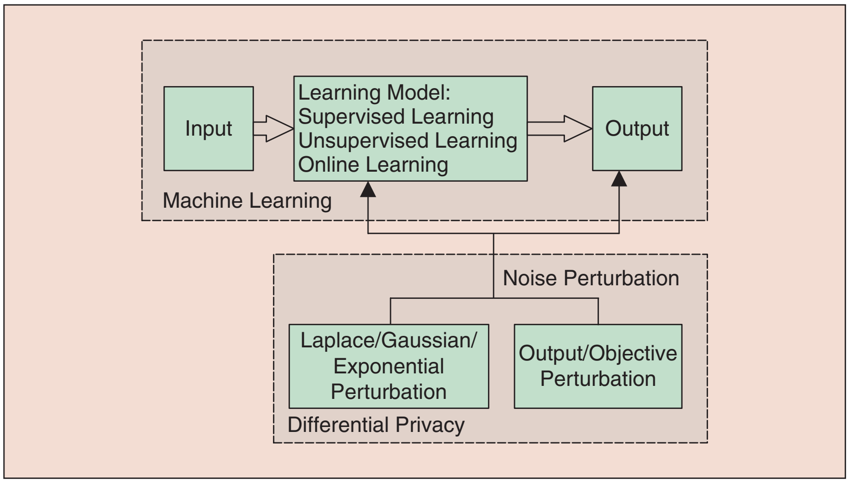

FIGURE 3 A classical learning framework with noise perturbation to satisfy differential privacy. Noise can be added into the learning model, the objective function, or the output results. In different cases, there are different methods to add noise perturbation.

To tackle the challenges in terms of model utility, privacy level and applications, several interesting directions deserve further exploration, such as devising alternative differential privacy mechanisms, enhancing the privacy and utility, presenting a unified private framework, incorporating differential privacy to other machine learning models, and involving novel distributed protocols.

numeric attributes, we need to derive the sensitivity of both mean and standard deviation due to the fact that the probability $P(X = \boldsymbol{x} \mid Y = c_j)$ depends on mean μ_j and variance σ_j^2. Assume that the values of attribute X_j lie in the range $[l_j, u_j]$, the sensitivity for the mean is $(u_j - l_j)/(n+1)$ and the sensitivity for the standard deviation is $\sqrt{n} \times (u_j - l_j)/(n+1)$. Following this, Laplace noise is added to parameters (counts for categorical attributes, mean and standard deviation for numeric attributes) to preserve the privacy.

Compared to the previous privacy-preserving algorithms that build the model over a single data provider, Li *et al.* [27] proposed a differentially private Naive Bayes algorithm that performs on multiple data sources. It aggregates the data of each owner and achieves privacy preserving during the training process without disclosing the privacy of each owner. In the first place, the cryptographic systems and the public parameters should be initialized. Then each data owner encrypts their own dataset and contributes it to the data collector. Some auxiliary information is also provided to the data collector. There is one more point that the data collector aggregates the encrypted data and adds the Laplace noise to the aggregated ciphertexts based on the auxiliary information to achieve privacy preserving. The last but not least, the data collector releases the differentially private Naive Bayes model to the data receiver. Furthermore, the proposed method can also preserve both statistics privacy and ownership privacy.

Decision tree learning. The learning process of a decision tree is an iterative process in which training data are segmented according to features which are selected recursively. A decision tree is constructed from the root that holds all training data. Then an optimal feature is chosen as per the information gain and it divides training data into subsets so that each subset has the best classification under current conditions.

Blum *et al.* [28] pioneered the first differentially private decision tree algorithm that performs on the Sub-Linear Queries (SuLQ) interface and preserves the privacy by adding noise to the information gain. The feature with the noisy information gain is chosen to partition a tree node when the noisy information gain of this feature is less than a specified threshold. However, the noisy information gain is evaluated separately for each feature in each iteration, which may result in a large amount of noise. What is more, the SuLQ fails to deal with continuous features. To overcome these disadvantages, Friedman *et al.* [29] employed the exponential mechanism in the step of feature selection so that not only continuous features can be tackled, but also less privacy budget is consumed than SuLQ. However, although the proposed method acquires a better performance than SuLQ, both of them still result in a large volume of noise.

To avoid the consumption of privacy budget during feature selection, Jagannathan *et al.* [30] proposed a differentially private random decision tree algorithm, which eliminates the pruning step by removing empty tree nodes and creating a tree in which all of the leaf nodes are at the same level. Furthermore, the leaf nodes of a random decision tree form a leaf vector, where the global sensitivity of the leaf vector is 1. The noise of $Lap(1/\epsilon)$ is added to each component of the leaf vector and the released noisy vector satisfies ϵ-differential privacy. The differentially private random decision tree can be generated from the noisy leaf vector. Unlike previous works that preserve the complete data distribution strictly, and in order to further reduce the consumption of noise, a differentially private random forest presented in [31] provides a more practical way to achieve data privacy preserving by only protecting the necessary statistics such as variance of the estimate, which can provide significantly higher utility.

Unlike the existing researches that focus on the scenarios where differential privacy is embedded in one-step data mining computation, Bai *et al.* [32] proposed an algorithm based on Markov Chain Monte Carlo (MCMC), which embeds differential privacy in a decision tree with different depths. To preserve privacy, the Laplace mechanism is applied for the generation of leaf node label and the exponential mechanism is applied for the split of a node. Furthermore, it is a polynomial-time algorithm for d-step computation embedding. In addition, a differentially private random decision forest algorithm is presented by Fletcher *et al.* [33], which preserves data privacy by using the exponential mechanism to output the class label rather than using a count query to return the class counts. It not only reduces the sensitivity of the query, but also achieves higher accuracy.

A straightforward implementation of differential privacy for decision trees often yields poor accuracy and stableness. To overcome this issue, a differentially private decision tree algorithm is proposed by Liu *et al.* [34], which preserves the privacy information based on a budget allocation strategy. In the strategy, the closer an internal node is to the root node, the smaller the budget allocated. In addition, the bagging technique [35] is used to construct ensemble models to avoid high variance and improve the classification performance through integrating decisions of multiple trees.

Boosting. Boosting is a family of algorithms that promote multiple weak learners to a strong learner. The working mechanism of boosting shares the following paradigm. The weight distribution of training data are first initialized. Then three computation steps are performed cyclically as follows. (1) Using a training dataset with the weight distribution D_m to obtain a basic classifier $G_m(\boldsymbol{x})$. (2) Calculating the classification error rate of $G_m(\boldsymbol{x})$ on the training dataset. (3) Updating the weight distribution D_{m+1} of the dataset. However, there are two considerable issues about boosting. One is how to change the weight or probability distribution of the training dataset in each round. The other is how to combine the weak learner into a strong learner.

Dwork *et al.* [36] designed a query-boosting algorithm, which aims to convert a weak and sometimes-accurate learner into a strong and accurate learner with differential privacy. It considers the input database as a training dataset, each row in the database as exactly a sample and almost does not compromise the accuracy. To achieve privacy preserving, it gradually changes the weight as a function of how accurate the answer is, rather than using a sharp threshold between the accurate and inaccurate answers, due to the fact that the change of each row in the database can affect the answers to all the queries and thus influence the distribution of queries.

Summary. The application of differential privacy in supervised learning algorithms represents an important branch of differentially private machine learning. Table I shown in the Supplementary Material summarizes and compares existing differentially private naive Bayes models and decision tree algorithms based on different mechanisms, advantages, disadvantages and privacy level. One of the advantages in differentially private supervised learning is that noise is easy to be added to satisfy differential privacy. For the differentially private decision tree, these algorithms are easy to retain good utility, but the privacy budget may be quickly consumed due to the continuous selection of split attributes, which requires a high privacy budget and may result in a bad privacy level.

2) Unsupervised Learning

Clustering. As the most basic clustering algorithm, *k*-means divides the unlabeled samples $D = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$ into k disjoint subsets $C = \{C_1, C_2, \ldots, C_k\}$, and the samples in the same subset are similar to each other. The goal of *k*-means clustering is to minimize $(1/n)\sum_{j=1}^{k}\sum_{\boldsymbol{x}\in C_i} \| \boldsymbol{x} - \boldsymbol{\mu}_{C_i} \|^2$ where $\boldsymbol{\mu}_{C_i}$ is the mean of samples in the subset C_i.

The differentially private clustering algorithm is performed by adding noise to each cluster center and the number of records in the center. However, it may result in a large sensitivity. To tackle this, Nissim *et al.* [23] used the local sensitivity instead of the global sensitivity to measure the sensitivity of cluster centers since the local sensitivity is much lower than the traditional global sensitivity. In addition, Feldman *et al.* [37] proposed a more restrict differentially private clustering algorithm, which defines the private coresets to preserve the privacy for *k*-mean queries. The coreset of a point set P is a small weighted set of records that captures geometric properties of these records. This algorithm satisfies differential privacy since the coreset is differentially private.

Due to the fact that the private clustering based on the sample-aggregate framework suffers from a poor utility in practice, Wang *et al.* [38] proposed a practical private subspace clustering algorithm based on the exponential mechanism. It preserves privacy information by acquiring the parameter $\theta = (\{S_l\}_{l=1}^{k}, \{C_i\}_{i=1}^{n})$ from the following distribution

$$p(\theta; X) \propto \exp\left(-\frac{\epsilon}{2}\sum_{i=1}^{n} d^2(\boldsymbol{x}^{(i)}, S_{C_i})\right), \tag{7}$$

where $S_l \in \mathbb{S}_d^q$ denotes the set of all q-dimensional subspace in $\mathbb{R}^d$, and $C_i \in \{1, \ldots, k\}$. To further improve the performance, Su *et al.* [39], [40] presented a noninteractive approach named Extended Uniform Grid *k*-Means (EUGkM), which publishes a differentially private synopsis for *k*-means clustering. Given a d-dimensional dataset, the dataset is divided into M equal-width grid cells. The differentially private *k*-means algorithm preserves data privacy by adding the Laplace noise to each cell count. When M is large, the noise will have greater impacts, and vice-versa. Schellekens *et al.* [41] proposed a differentially private compressive *k*-means method, which provides privacy preserving by combining the Laplace noise with subsampling. This method performs at least as well as previously developed ones while requiring fewer computations.

Dimensionality reduction. Dimensionality reduction transforms the original high-dimensional attribute space into a low-dimensional subspace through mathematical transformations. In this subspace, the sample density is greatly improved and the distance calculation becomes easier. Principal Component Analysis (PCA) is a popular method of dimensionality reduction that aims to calculate k new irrelevant attributes ranked by the importance from large to small, which is a linear combination of original attributes. For a set of n vectors $D = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$, where each $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ corresponds to the privacy data of one individual. Suppose $X = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}]$ is the matrix whose columns are data vectors $\{\boldsymbol{x}^{(i)}\}$. The positive semidefinite matrix $A = (1/n)XX^T$ denotes the $d \times d$ second-moment matrix of the data. Let the eigenvalues of A be $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_d(A) \geq 0$ and Λ be a $d \times d$ diagonal matrix with $\Lambda_{ii} = \lambda_i(A)$. The Singular Value Decomposition (SVD) gives $A = V^T\Lambda V$, where V is an orthonormal matrix of eigenvectors.

The Sub-Linear Queries [28] method adds noise to the second moment matrix and runs PCA on the noisy matrix, which may impact the quality of approximation. Unlike the Sub-Linear Queries method, Chaudhuri *et al.* [42] proposed PPCA that explicitly considers the quality of approximation and the sample complexity of it scales as $O(d)$. The PPCA algorithm preserves the privacy information by randomly sampling a *k*-dimensional subspace from the matrix Bingham distribution $\mathrm{BMF}_k(B)$ [43] based on the exponential mechanism. The density of the matrix Bingham distribution is equals to

$$f(V) = \frac{1}{{}_1F_1\left(\frac{1}{2}k, \frac{1}{2}d, B\right)} \exp(\mathrm{tr}(V^T B V)), \tag{8}$$

where V is a $d \times k$ matrix whose columns are orthonormal and ${}_1F_1((1/2)k, (1/2)d, B)$ is a confluent hypergeometric function. Kapralov et al. [44] presented a low rank approximation algorithm based on the Laplace and exponential mechanisms, which provides a strict guarantee on convergence. However, it is complicated and takes $O(d^6/\epsilon)$ time complexity.

To acquire a better utility guarantee, Jiang *et al.* [45] proposed a differentially private PCA approach that uses a Wishart distribution to generate noise. Unlike traditional methods that add noise when computing the top-k subspace of A, the proposed mechanism generates a noisy sample covariance matrix before computing the eigenspace. The magnitude of the noise matrix directly determines how large the effects on the original matrix are. This method takes $O(kd^2)$ running time. Most of differentially private PCA algorithms either employ the computationally intensive exponential mechanism or require an access to the covariance matrix. Both of them fail to utilize the potential sparsity of the data. To overcome these issues, a differentially private PCA mechanism based on the smooth sensitivity is presented in [46], which preserves privacy by employing output perturbation. Moreover, a post processing step is conducted so that there is reasonable noise added to the output.

Plenty of machine learning tasks perform on the datasets that contain sensitive information and hold at different locations. The differentially private algorithms perform worse in the distributed environment since the introduction of a larger volume of noise. However, a distributed differentially private algorithm for PCA is proposed in [47], which employs a correlated noise design scheme to alleviate the effects of noise and achieves the same noise level as the centralized scenario. This method defines a noise generator to generate the $D \times D$ matrix E_s i.i.d. $\sim\mathcal{N}(0,\sigma_e^2)$. In addition, an aggregator generates the $D \times D$ matrix F_s i.i.d. $\sim\mathcal{N}(0,\sigma_f^2)$ and the sites generate the $D \times D$ matrix G_s i.i.d. $\sim\mathcal{N}(0,\sigma_g^2)$. The sites that hold a smaller number of samples preserve privacy by using the $D \times D$ matrix sent from the random noise generator and the aggregator to compute the noisy estimate of the local second-moment matrix by

$$\hat{A}_s \leftarrow A_s + E_s + F_s + G_s, \tag{9}$$

where $A_s = (1/n) X_s X_s^T$. Finally, $\hat{A}_s$ is sent to the aggregator.

Summary. Differentially private unsupervised learning has made great developments in recent years. We summarize and compare differentially private algorithms applied in clustering and dimensionality reduction in Table II of the Supplementary Material. Differentially private clustering algorithms usually retain good utility yet come along with high sensitivity. Local sensitivity and smooth sensitivity can be used to solve this problem to a certain extent. Most of differentially private PCA algorithms rely on a large number of computations, which require high time complexity and fail to utilize the potential sparsity of the data. Some techniques are designed to reduce time complexity and there remains room to be further improved. In addition, a distributed differentially private algorithm [47] for PCA is developed.

3) Online Learning

Online learning is an outstanding method of machine learning in which data become available in a sequential order, and is used to update the best predictor for future data at each step rather than generate the best predictor by learning on the entire training dataset at once. In online learning, the online convex programming (OCP) solves convex programming problems in an online manner, which maps a function sequence $F = \langle f_1, f_2, \ldots, f_T \rangle$ to a sequence of points $\boldsymbol{x} = \langle \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)}, \ldots, \boldsymbol{x}^{(T+1)} \rangle$. The goal of it is to minimize the regret as

$$\mathcal{R}(T) = \sum_{t=1}^{T} f_t(\boldsymbol{x}^{(t)}) - \min_{\boldsymbol{x}^* \in C} \sum_{t=1}^{T} f_t(\boldsymbol{x}^*). \tag{10}$$

Jain *et al.* [48] proposed a differentially private framework for solving OCP problems, which is instantiated by Implicit Gradient Decent (IGD) [49] and Generalized Infinitesimal Gradient Ascent (GIGA) [50]. Assume that OCP satisfies the L_2-sensitivity and the regret bound $\mathcal{R}(T)$, the framework provides $\tilde{O}(\sqrt{T})$ regret and preserves privacy by adding noise to each sample $\boldsymbol{x}^{(t)}$ and using the perturbed sample for the future computation. For the problem of online linear optimization in the full information and bandit settings with optimal $\tilde{O}(\sqrt{T})$ regret bounds, a differentially private algorithm is presented in [51], which preserves privacy by ensuring loss vectors of the entire sequence are differentially private.

Li *et al.* [52] developed a much faster differentially private distributed online learning algorithm (DOLA) on the data collected from distributed data sources, it can be also used for high-dimensional data optimization. Both ϵ and (ϵ, δ)-differential privacy are provided for DOLA, and private regret bounds have the same order of $O(\sqrt{T})$ and $O(\log T)$ respectively with the non-private algorithm. Specifically, the ϵ-differentially private DOLA is performed by adding the Laplace noise to the learnable parameter ω_{t+1}^{i}, and the Laplace output perturbation is broadcasted to neighbors $\mathcal{G}(t+1)_i$. In addition, the (ϵ, δ)-differentially private DOLA is performed by adding the Gaussian noise to the updated parameter ω and broadcasting the Gaussian output perturbation to neighbors $\mathcal{G}(t)_i$. Both of them provide a strong privacy guarantee for individuals.

B. Output/Objective Perturbation Mechanism

The output and objective perturbation mechanisms are two generic differentially private methods to achieve privacy preserving. The output perturbation mechanism is performed by adding an amount of noise to the model output while the objective perturbation mechanism can be implemented by adding noise to the objective function and optimizing the perturbed objective function. There is no doubt that both of them play an indispensable role in differentially private machine learning algorithms, and can be used in empirical risk minimization (ERM) and distributed optimization.

1) Differentially Private ERM

The empirical risk is the average loss of the machine learning model on the training dataset D. The objective of ERM is to obtain the optimal parameter ω^* that minimizes the empirical risk as

$$\min_{\omega} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(\omega, \boldsymbol{x}_i); y_i), \tag{11}$$

where f is the prediction function with the parameter ω. Differentially private ERM provides us an opportunity to preserve privacy in a dataset effectively given a target excess risk. The utility of differentially private ERM can be measured by the risk bound, which is related to the dimension and the size of the dataset. There are quite a few differentially private ERM methods based on output or objective perturbation, including linear regression, logistic regression, linear SVM and kernel SVM, all of which are supervised models.

Linear regression. Suppose a dataset D contains n tuples $\{t^{(1)}, \ldots, t^{(n)}\}$, d explanatory attributes $X_1, X_2, \ldots, X_d$ and one response attribute Y, where each tuple $t^{(i)} = (\boldsymbol{x}^{(i)}, y^{(i)})$ and $\boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$. Linear regression aims to learn a model to predict Y given $X_1, X_2, \ldots, X_d$. The cost function $f(t^{(i)}, \omega) = (y^{(i)} - \omega^T \boldsymbol{x}^{(i)})^2$ is used to measure the difference between the true and predicted values of y_i. The optimal model parameter is defined as $\omega *= \operatorname*{argmin}_{\omega} \sum_{i=1}^{n} f(t^{(i)}, \omega)$.

A differentially private linear regression model suitable for low-dimensional datasets is presented in [53]. It preserves the data privacy by adding noise to histograms of the input data and producing a synthetic dataset based on the perturbed histogram. However, the existing works are limited to nonstandard types of regression analysis or unable to output the regression results as accurate as possible. Therefore, Zhang *et al.* [54] proposed the functional mechanism, which achieves the preservation of sensitive data by perturbing the objective function $f(t^{(i)}, \omega)$ and releasing the model parameters to minimize the perturbed objective function. Specifically, the cost function $f(t^{(i)}, \omega)$ can be written as a polynomial of $\omega_1, \ldots, \omega_d$, i.e., for some $J \in [0, \infty]$, we have

$$f(t^{(i)}, \omega) = \sum_{j=0}^{J} \sum_{\phi \in \Phi_j} \lambda_{\phi t^{(i)}} \phi(\omega), \tag{12}$$

where Φ_j denotes a set of products of $\omega_1, \ldots, \omega_d$ with degree j and $\lambda_{\phi t^{(i)}}$ denotes the coefficient of $\phi(\omega)$. The objective function is perturbed by adding noise to coefficients $\lambda_{\phi t^{(i)}}$.

Compared to traditional methods that subject to a fixed privacy budget ϵ, a general noise reduction framework that has a better performance for providing privacy preserving on regularized linear regression is presented in [55], which takes a set of privacy levels $\epsilon_1 < \cdots < \epsilon_T$ as input and outputs a sequence of hypotheses $\theta^1, \ldots, \theta^T$, and each θ^t satisfies ϵ_t-differential privacy. The optimal solution θ^* is

$$\begin{aligned} \theta^* &= \operatorname*{argmin}_{\theta} \mathcal{L}(\theta, D) \\ &= \operatorname*{argmin}_{\theta} \frac{(\theta^T (\boldsymbol{x}^T \boldsymbol{x}) \theta - 2 \langle \boldsymbol{x}^T y, \theta \rangle)}{2n} \\ &\quad + \frac{\lambda \| \theta \|_2^2}{2}, \end{aligned} \tag{13}$$

where λ is a regularization parameter. The proposed noise reduction framework preserves the data privacy by adding the Laplace noise to each entry of $\boldsymbol{x}^T \boldsymbol{x}$ and $\boldsymbol{x}^T y$ based on the covariance perturbation. Finally, the private hypothesis is derived by solving the noisy version of the optimization problem, and the algorithm halts when the released hypothesis θ^t achieves the accuracy goal.

Logistic regression. As a binary classification model, the logistic regression model performed on the dataset $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(\mathrm{n})}, y^{(\mathrm{n})})\}$ predicts $y^{(i)} = 1$ with the probability $\exp(\omega^{*T} \boldsymbol{x}^{(i)}) / (1 + \exp(\omega^{*T} \boldsymbol{x}^{(i)}))$, where ω^* is a d-dimensional real vector that minimizes the cost function $f(t^{(i)}, \omega) = \log(1 + \exp(\omega^T \boldsymbol{x}^{(i)})) - y^{(i)} \omega^T \boldsymbol{x}^{(i)}$. In general, a regularization term $(\lambda / 2m) \|\omega\|_2^2$ is added to the cost function to prevent overfitting, where $\lambda > 0$ is a hyperparameter of the cost function.

Chaudhuri *et al.* [18], [56] demonstrated that regularized logistic regression can be combined with differential privacy based on output or objective perturbation directly, since the cost function of regularized logistic regression is continuous, differentiable and doubly differentiable. However, the proposed algorithm is inapplicable for standard logistic regression. Besides, the regression model proposed by Jing [53] can also preserve the privacy by perturbing histograms, which avoids conducting a sensitivity analysis directly on regression outputs, but it is limited to datasets with small dimensionality or it leads to a poor accuracy.

The differentially private functional perturbation mechanism [54] is proposed to solve the above problems. However, the polynomial form of the objective function of logistic regression contains terms with unbounded degrees, so that we need to derive an approximate polynomial form of the objective function based on Taylor expansions. Assume that there exist $2m$ functions $f_1, \ldots, f_m$ and $g_1, \ldots, g_m$, such that $f(t^{(i)}, \omega) = \sum_{l=1}^{m} f_l(g_l(t^{(i)}, \omega))$ and each g_l is a polynomial function of $\omega_1, \ldots, \omega_d$. Given the above decomposition of $f(t^{(i)}, \omega)$, the Taylor expansion can be applied on each $f_l(\cdot)$ to obtain the following equation

$$\begin{aligned} &\hat{f}(t^{(i)}, \omega) \\ &\quad = \sum_{l=1}^{m} \sum_{k=0}^{\infty} \frac{f_l^{(k)}(z_l)}{k!} (g_l(t^{(i)}, \omega) - z_l)^k, \end{aligned} \tag{14}$$

where z_l is a real number. The acquired objective function $\hat{f}(t_i, \omega)$ can be perturbed by adding noise to its coefficients.

Data owners may have different privacy preferences and the same privacy preserving provided for all individuals will limit the accuracy of the model. Therefore, Li *et al.* [57] proposed the privacy-aware mechanism and utility-based partitioning mechanism to acquire a better performance. The former is to minimize the waste of privacy budget whereas the latter is to maximize the utility for a given aggregate analysis. Given the minimum threshold T of partitions size n_i for the differentially private mechanism, the privacy-aware partitioning takes $O(mn\log n)$ complexity where $m = n/T$ and the utility-based partitioning takes $O(n)$.

Linear SVM. Linear SVM is a linear classifier which constructs a model from the dataset $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})\}$ and finds the optimal dividing hyperplane $\omega^* \boldsymbol{x} + b = 0$ to separate samples into different categories. The parameter ω is calculated by minimizing the loss function $\mathcal{L}_{SVM}(\omega) = \min_{\omega,b}(1/2)\|\omega\|^2 + C\sum_{i=1}^{m} \max(0, 1 - y_i(\omega^T \boldsymbol{x}^{(i)} + b))$, where C is a constant and $C > 0$.

Chaudhuri *et al.* [18], [56] demonstrated that linear SVM can be combined with the differential privacy mechanism by applying output or objective perturbation to preserve personal data on condition that loss functions are differentiable and convex. However, the loss function of linear SVM does not satisfy the conditions, since it is continuous but not differentiable. There are two alternative solutions to tackle it. One solution is to approximate $\mathcal{L}_{SVM}$ by a piecewise loss function and another is to use the Huber loss. After the transformation of the loss function, the differentially private linear SVM can be performed based on output or objective perturbation.

Kernel SVM. Linear SVM is an effective method to solve the linear classification problem, but can not solve nonlinear classification problems. We can apply kernel SVM that involves the kernel tricks to address the nonlinear classification problem. Kernel SVM aims to transform the input space to a linearly separable feature space through a nonlinear transformation, so that the classification problem can be accomplished by solving the linear SVM in the feature space. The kernel SVM predicts the label $Y = \text{sign}(\sum_{i=1}^{m} \alpha_i \kappa(X, \boldsymbol{x}^{(i)}))$ when classifying a sample with feature X, where the Lagrange multiplier $\alpha_i \geq 0$ and $\kappa(X, \boldsymbol{x}^{(i)})$ is the kernel function.

The algorithm proposed in [58] maps the input data to a randomized low-dimensional feature space and applies the existing differentially private linear methods to preserve privacy. The presented mechanism indicates that inner products of the transformed data are approximately equal to those in the feature space of the specified shift-invariant kernel. Thus, a kernel SVM model is transformed into a linear SVM model and differentially private linear SVM mechanisms can be used in the kernel SVM model to preserve the individual privacy. Rubinstein *et al.* [59] presented an efficient mechanism for potentially infinite-dimensional feature mappings with translation-invariant kernels, which minimizes the regularized empirical risk in a random Reproducing Kernel Hilbert Space (RKHS) $\hat{\mathcal{H}}$ whose kernel uniformly approximates the desired kernel $\mathcal{H}$ with high probability. The proposed method preserves the privacy of individual entries by adding appropriate Laplace noise to the weight vector $\tilde{\omega}$. Another work for differentially private kernel SVM is proposed in [56], which also uses an approximation method [58] to approximate the kernel function based on random projections and it projects data from the original sample space to a space independent of private training data. This differentially private kernel SVM is performed by transforming nonlinear classification to linear classification and perturbs the objective function based on output or objective perturbation.

However, the above algorithms are always restricted to the specific translation-invariant kernels and may be not suitable for various kernels. Therefore, Jain *et al.* [60] presented a differentially private kernel SVM algorithm for all RKHS kernels. The user sends a small subset of test data to the trusted learner and the trusted learner returns a differentially private version $\hat{\omega}$ of the optimal solution to the kernel ERM (kERM) (ω^*) over T rounds. The proposed test data independent learner is proved to satisfy (ϵ, δ)-differential privacy and the sample complexity is $O(d^{1/3})$ compared to $O(d)$ for [56], [59], where d represents the dimension.

Summary. Output and objective perturbation are common techniques in differential private empirical risk minimization. Objective perturbation performs better than output perturbation because additional noise added on the objective function will not significantly affect the performance, while that added on the output does. However, objective perturbation is premised on the fact that the objective function must be convex, differential, or satisfy other conditions. More specific comparisons are shown in Table III of the Supplementary Material.

2) Differentially Private Distributed Optimization

The distributed optimization is a crucial approach to tackle machine learning problems that can be transformed into an objective function in a distributed manner. Given a group of n agents, each agent $i \in \{1, \ldots, n\}$ has its own corresponding objective function $f_i : D \to \mathbb{R}$ where $D \subset \mathbb{R}^n$. The objective function f_i is only known to the agent i, which is convex and twice continuously differentiable. The goal of distributed optimization problems for all the agents is

$$\min_{\boldsymbol{x} \in D} f(\boldsymbol{x}) = \sum_i f_i(\boldsymbol{x}), \tag{15}$$

which subjects to $\begin{cases} A\boldsymbol{x} = b \\ H(\boldsymbol{x}) \leq 0 \end{cases}$, where $A \in \mathbb{R}^{s \times d}, b \in \mathbb{R}^s$ and the component functions of $H : D \to \mathbb{R}^m$ are convex. Differentially private distributed optimization aims to minimize the sum of individual objective functions while preventing privacy disclosure based on output or objective perturbation.

For preserving privacy in distributed optimization, it is indispensable for us to avoid the information inference about the objective function when agents exchange the messages. Huang *et al.* [61] proposed a private distributed optimization problem (PDOP) based on objective perturbation, which effectively provides privacy preserving for the objective function of each agent. Any change in the objective function only leads to nonsubstantial transformations in message statistics. The PDOP tackles privacy disclosure in objective functions by adding an amount of Laplace noise to the optimal point that has been estimated and broadcasted the noisy estimate to its adjacent agents. In addition, the sensitivity $S(f)$ of PDOP is a key parameter that determines the amount of noise should be added. Therefore, it is imperative to keep the Laplace noise

$Lap(S(f)/\epsilon)$ smaller to balance privacy and utility. Moreover, the accuracy of PDOP has the order of $O(1/\epsilon^2)$.

However, the output of the algorithm is inconsistent with the true optimizer for any fixed level of privacy. Therefore, Nozari *et al.* [62] proposed a differentially private distributed convex optimization framework, in which each agent perturbs the objective function that belongs to itself based on functional perturbation. The presented method uses the trusty distributed coordination algorithm to optimize the sum of noisy objective functions. Finally, the distance between the noisy optimizer and the expected optimizer is bounded explicitly so that the accuracy of the presented model can be guaranteed.

C. Sample Complexity

The sample complexity shows lower bounds on the numbers of samples to reach a particular accuracy for a learning model. It is strongly related to the theory of probably approximately correct (PAC) learning and the VC dimension. For a learning model, the concept is defined as the mapping from the sample space $\mathcal{X}$ to the output space $\mathcal{Y}$. For any sample $(\boldsymbol{x}, y)$, if the mapping $c(\boldsymbol{x})$ is equal to y, c is viewed as the target concept. Concept class $\mathcal{C}$ is the set of all target concepts. A hypothesis space $\mathcal{H}$ represents the set of all the possible concepts and a concept in the hypothesis space is called a hypothesis h. It is called properly PAC learnable provided that $\mathcal{H} = \mathcal{C}$ holds in a learning model [63].

The sample complexity of a non-private learning model that is efficiently PAC learnable only requires a constant number of samples. Blum *et al.* [64] demonstrated that to satisfy ϵ-differential privacy, the lowest sample complexity of a privacy learning model is $O(\log|\mathcal{X}| \cdot VC(\mathcal{C}))$, which is higher than the non-private learning model. To ensure both privacy and utility of privacy learning models, researchers conducted several research works on reducing the sample complexity [65].

Replacing the proper learner by the improper learner can also reduce the sample complexity in privacy learning. Beimel *et al.* [66] demonstrated that for a proper private learner, the sample complexity can be approximated as $\Omega(d)$ where d is the dimension of concept class. However if the learner is improper, the sample complexity can be reduced to $O(\log(d))$. The improper learner can also be built upon a probabilistic representation of $\mathcal{C}$. A list of collections $\{\mathcal{H}_1, \ldots, \mathcal{H}_r\}$, rather than one collection $\mathcal{H}$, is utilized to represent $\mathcal{C}$. If the sampled collection $\mathcal{H}_i$ belongs to the list, there will be a hypothesis $h \in \mathcal{H}$ that is close to c with high probability. The learning model samples $\mathcal{H}_i$ from the list and then selects a hypothesis from $\mathcal{H}_i$ based on the exponential mechanism. The sample complexity can be reduced to $O(\max_i \ln|\mathcal{H}_i|)$. However, the workload of this method may increase exponentially. For a private learning model with constant sample complexity, if $\mathcal{H} = \mathcal{C}$, the time for evaluation is also a constant. However if $\mathcal{H} \neq \mathcal{C}$, it needs $O(\exp(d))$ time for evaluation [67].

There is another feasible way to reduce the sample complexity by relaxing the privacy requirement. Chaudhuri *et al.* [68] assumed that with label privacy guarantee rather than all attributes of samples, at least $\Omega(d')$ samples are required for a given hypothesis set in any learning algorithm, where d' is the doubling dimension of the disagreement metric. However, this method only applies to samples whose other attributes are nonsensitive except for labels. Otherwise, there still exists the risk of leaking sensitive information.

Inspired by semi-supervised learning, Beimel *et al.* [69] designed a novel technique for reducing the labeled sample complexity of a given private learning model. It sanitizes unlabeled data to create a synthetic dataset, and chooses a subset of the hypotheses of size $2^{O(\mathrm{VC}(\mathcal{C}))}$ based on $O(\mathrm{VC}(\mathcal{C}))$ labeled examples. The complexity of labeled samples is $O(\mathrm{VC}(\mathcal{C}))$ while the complexity of unlabeled samples is $O(d \cdot \mathrm{VC}(\mathcal{C}))$. The high sample complexity is unavoidable for unlabeled samples in any generic ϵ-differentially private learning models.

Extending ϵ-differential privacy to (ϵ, δ)-differential privacy is another method to reduce the sample complexity. Suppose the family of point functions $\{c_i(x) = 1\}$ iff $x = i$ otherwise $\{c_i(x) = 0\}$, and the family of threshold functions $\{c_i(x) = 1\}$ iff $x \leq i$. For (ϵ, δ)-differentially private and properly PAC learnable point functions, the sample complexity is a constant, while the threshold functions require $2^{O(\log^* d)}$ [70]. Bun *et al.* [71] indicated that (ϵ, δ)-differentially private and properly learning threshold functions require $\Omega(\log^* d)$ samples. However, these works are based on relatively simple concept classes, and have not yet been extended to more complex concept classes.

Summary. PAC learning theory is used to analyze the relationship between the learning model and samples it requires to reach a certain accuracy. It reveals that the sample complexity of privacy learning is higher than that in non-privacy learning. So many endeavors have been devoted to reducing the sample complexity of privacy learning. However, most of them focus on theoretical research and have few practical application. Table IV shown in the Supplementary Material summarizes the complexity, advantages, disadvantages and privacy level of those methods.

IV. Differentially Private Deep Learning

Besides shallow machine learning models, deep learning with differential privacy is another popular area. In this section, we present a taxonomy of recent differentially private deep learning methods according to different noise perturbation mechanisms. For each setting, we explain its characteristics and introduce the representative approaches in detail.

A. Laplace/Gaussian Mechanism

In fact, the Laplace or Gaussian mechanism can be incorporated into existing deep learning models to further enhance the privacy. There are generally two types of possible solutions. One is to introduce the Laplace or Gaussian mechanism into general training algorithms for deep learning including centralized and distributed algorithms. Another is to design a customized private version of the specific deep learning models such as long short-term memory networks (LSTM), generative neural networks (GAN) and their variants, etc.

1) Centralized Training Algorithms

Deep learning models are often optimized by gradient descent or its variants through which we can minimize the nonlinear objective function and find optimal model parameters. In most cases, the training data are held centrally, which we called centralized training. In an early work, stochastic gradient descent with differentially private updates has been derived for general convex objectives [72]. Inspired by this, an intuitive method is adding random noise to gradients for privacy preserving.

NoisySGD, a differentially private version of the SGD algorithm, is proposed in [73] for preserving privacy of training data through differentially private optimization. Based on the basic Laplace mechanism, NoisySGD clips each gradient by l_2 norm, groups the batches into a lot and adds noise to the sum of gradients of each lot, working on the training process of non-convex deep learning models. Further, it refines the privacy loss analysis by the modest moments accountant. It can control the effect of training data over the course of SGD computation and output a privacy-preserving deep learning model. Following the learning architecture of NoisySGD, other differentially private deep learning techniques based on gradient perturbation have been proposed in [74]–[80].

In NoisySGD [73], the amount of random noise and the privacy budget remain increasing with the increase in the number of training epochs, which is not expected since privacy budget is usually limited. Besides, the amount of noise remains unchanged regardless of the importance of different parameters in existing differentially private deep learning techniques. The work of Shokri *et al.* [81], also suffers the same problems. To tackle these drawbacks, Phan *et al.* [82] proposed a highly effective mechanism for differential privacy preservation in deep learning. Laplace noise is added to the affine transformations of neurons and the loss functions only once. The input features are adaptively perturbed according to the contribution of different features upon the model output. In addition, it can be applied in various deep models as NoisySGD [73].

2) Distributed Training Algorithms

A prerequisite for centralized training algorithms is the massive data available for training the deep learning model. Nevertheless, an institution only owns limited amount of data in general, which may result in overfitting while training deep learning models. Even more, a crowdsourced data collection suffers from obvious privacy issues because data owners can neither delete nor restrict the purpose once collected. Recently, there emerges a paradigm of distributed deep learning where multiple participants jointly train a deep learning model through a central server to achieve common objectives without sharing the private data.

Shokri *et al.* [81] presented a pioneer work of incorporating differential privacy into distributed deep learning. They carefully designed a practical training framework that enables multiple participants to collaboratively learn a desirable deep learning model without sharing their own training data. Under the assumption that different participants have the same objective function in advance, this framework is optimized by the proposed distributed selective SGD protocol. In this protocol, each participant independently trains their local model on their own dataset and only asynchronously uploads part of truncated and perturbed gradients, under a consistent differential privacy mechanism. Each participant can download latest parameters shared by other participants to enhance its own local model. This protocol explicitly avoids the leakage of the sensitive information and the empirical evaluation proves that it can actually achieve comparable accuracy to conventional SGD.

Zhang *et al.* [83] proposed another method for privacy preserving in multiparty deep learning, whose scenario is similar to that in [81]. For the reason that each party operates under the local private context, the injected randomization is often overly conservative, resulting in great uncertainty of information disclosure and significant utility loss in the global model. To solve this issue, this method not only enforces differentially private randomization to local gradients in local models but also considers ρ-visibility for obtaining more secure aggregation of local gradients based on homomorphic encryption and threshold secret sharing. Through the synergy of multi-participants, it can provide us with powerful privacy assurance and high effectiveness simultaneously.

Following the privacy-preserving training process of deep neural networks in NoisySGD, Chase *et al.* [74] married differential privacy with secure multiparty computation to avoid the privacy leakage in collaborative machine learning. They designed a protocol of training collaborative neural networks, in which the private gradient descent method adds random noise from an appropriate distribution to gradients. Then the collaborative private gradient descent method ensures that the compounded information per mini-batch will not be disclosed too much with the increase of the number of participants.

Different from the above differentially private distributed training frameworks [74], [81], [83] that add random noise to gradients, Papernot *et al.* [84] introduced Private Aggregation of Teacher Ensembles (PATE) for learning generally applicable privacy-preserving models from disjoint private data, agnostic to model details and optimization algorithms. Different data owners with the same machine learning tasks train their own teacher model on disjoint sensitive data independently. The votes of multiple teacher models can be aggregated and the Laplace noise is added to aggregation results:

$$f(\boldsymbol{x}) = \operatorname*{argmax}_{j}\left\{n_j(\boldsymbol{x}) + Lap\left(\frac{1}{\gamma}\right)\right\}, \quad (16)$$

where γ is the privacy parameter and $n_j(\boldsymbol{x})$ is the number of teachers that assign class j to the input $\boldsymbol{x}$. Then, a student model can be trained on unlabeled public data by semi-supervised knowledge transfer from the teacher ensemble, providing a releasable model that will not expose privacy in the original data.

However, the original PATE [84] is only practical on simple classification tasks since it requires more queries and has a large privacy cost when handling class imbalance problems. In fact, data irregularities are ubiquitous in pattern classification [85]. Scalable private learning with PATE, named SPATE [86], is a novel aggregation mechanism that is more selective and adds less noise. If the teacher's answers are not consistent, the answer can be omitted. The teachers can also give no answers when the student can confidently obtain the right answer. To involve less noise for preserving privacy in the aggregation of votes, SPATE adds Gaussian noise instead of Laplace noise, which is beneficial for the practicality of prediction tasks with many output classes.

3) Long Short-Term Memory Networks

As a standard language model, long short-term memory networks (LSTM) can capture long term dependencies and have been successfully applied to speech recognition, machine translation, and sentiment analysis, etc. Unfortunately, these training data are all privacy sensitive. Considering the fact that each user may contribute a multitude of training samples in language modeling, the approach [75] provides user-level privacy guarantees in LSTM instead of example-level privacy in prior works. In spite of the complex internal structure, McMahan *et al.* designed a model based on federated learning for next-word prediction in mobile keyboards. The user-level privacy preserving can be achieved in federated algorithms by adding Gaussian noise and enforcing clipping each-user update in the iterative training.

4) Generative Neural Networks

Generative models, training on realistic data collected from users, are widely used for generating various desirable data. Generative neural networks [87] iteratively train the generator and discriminator until the generated sample is indistinguishable for the discriminator from the true sample. Releasing of differentially private data faces many challenges in preserving inherent correlation structure of various types of data. To resolve these challenges, Lu *et al.* [77] presented a unified framework of generating and publishing synthetic tabular or graph data through modeling the input distribution based on generative adversarial networks trained by NoisySGD [73]. To tackle the data scarcity problem in some domains, Xie *et al.* [80] presented another unified framework of training generative adversarial networks in a differentially private manner. During the training procedure, carefully designed noise is added on gradients of the discriminator and iterative gradient descent is enforced with gradient clipping.

In medical practice, clinical data are extremely scarce and often need to be shared. To address the challenges in clinical data sharing, the method in [78] trains auxiliary classifier generative adversarial networks with differential privacy for generating shareable and reanalysis biomedical data. For preserving the privacy of participants, it clips the norm of gradients and adds Gaussian noise while training the discriminator.

Instead of adding noise to gradients of the discriminator during training in [78], [80], PATE-GAN [88] modifies PATE [84] and integrates it to GANs for generating synthetic data in a differentially private manner. In order to build a differentially private generative model, PATE-GAN designs multiple teacher distinguishers and a student distinguisher. Multiple teacher distinguishers have access to the real data that are partitioned in advance, and the differentially private aggregation of their outputs are used to train the student distinguisher. The resulting model can be applicable for generating synthetic data while providing rigorous privacy guarantees for the original dataset.

Different from previous studies [77], [78], [80] that release the synthetic data, the approach proposed in [79] aims at publishing a differentially private deep generative model that is trained on original data. Within the training of generative adversarial networks, it clips the norm of gradients and adds appropriate Gaussian noise while updating the discriminator that is directly accessible to original data. To further improve the training stability and convergence rate, three optimization strategies are proposed. Thus, analysts can use the well-trained private generative model to produce high-quality data.

In [89], GANobfuscator, a differentially private generative adversarial network is proposed to mitigate information leakage under GAN. Specifically, the carefully designed noise is added to gradients to achieve differential privacy within the learning procedure, and the gradient pruning is presented to enhance the privacy and improve the stability and scalability of generative model training itself. Unlike the privacy-preserving framework mentioned in PATE [84], whose privacy loss is proportional to the amount of data needed to be labeled in the public dataset, the privacy loss of GANobfuscator is independent of the amount of generated data, which enables GANobfuscator for a wide variety of real-world scenarios.

Note that releasing generative models trained by standard SGD techniques may leak the privacy of users. In [76], data are separated to different clusters by differentially private kernel k-means and separate generative neural models are trained on each cluster by improved differentially private gradient descent. As a result, we can learn private generative models that provide differential privacy for each individual in the training data and generate realistic synthetic samples.

5) Summary

In Table V of the Supplementary Material, we summarize and compare the aforementioned differentially private deep learning methods based on the Laplace or Gaussian mechanism, mainly according to specific mechanisms, advantages, disadvantages and privacy level. To achieve differential privacy, the Laplace or Gaussian mechanism is usually performed upon gradient during model training, because the optimization of neural network based deep learning methods is all based on gradient descent. Remarkably, generating sharable and non-private data are a paramount practical application of differentially private deep learning.

B. Output/Objective Perturbation Mechanism

This section reveals the incorporation of the output or objective perturbation mechanism for privacy preserving in deep learning models including deep auto-encoders (AE), convolutional deep belief networks (CDBN) and a network embedding method named DeepWalk. The existing three research works are all based on objective perturbation.

1) Deep Auto-Encoders

Deep auto-encoders [90] are composed of multiple auto-encoders and can be trained for extracting useful latent features in an unsupervised manner. As the first attempt of applying objective perturbation for privacy preserving in deep learning models, Phan *et al.* [91] tried to ensure that the adversaries will not learn any sensitive information from deep auto-encoders even if they possess all the remaining tuples of the sensitive data. It enforces polynomial approximation of the data reconstruction and the cross-entropy objective function by Taylor expansion. Then it adds appropriate noise into coefficients of the polynomial approximation. After training deep auto-encoders by gradient descent methods, the optimal parameters can be obtained, which do not disclose any private information of the training data when the model is released.

2) Convolutional Deep Belief Networks

Convolutional deep belief networks (CDBNs) represent one of the well-known hierarchical generative models, which consist of multiple convolutional restricted Boltzmann machines [92]. Analogous to [91], Phan *et al.* [93] introduced differential privacy to CDBNs. Realizing the fact that the objective function of CDBNs is more complicated than that of auto-encoders, private CDBNs ingeniously utilize the Chebyshev expansion to obtain the polynomial approximation of the energy-based objective functions of CDBNs. To achieve differential privacy, coefficients of the polynomial representation are perturbed by adding random noise. Furthermore, a single-layer private CDBN can be stacked to deep private CDBNs.

3) DeepWalk

As a new learning paradigm for network analysis, network embedding encodes nodes in a network into a low-dimensional vector space, and simultaneously characterizes structure information of a network. Unfortunately, almost all of the existing network embedding methods ignore the risk of releasing nodes representations, which may cause link privacy leakage for nodes in a network. DPNE [94] develops a differentially private version of DeepWalk [95] that is revealed to be equivalent to factorization of a normalized Laplacian matrix in [96]. By perturbing the objective function of matrix factorization, the embedding representations learned by DPNE can achieve differential privacy. In addition, DPNE can be applied in other network embedding methods that can be equal to factorize a certain matrix.

4) Summary

There are only three related works conducted on output or objective perturbation. The reason is that the objective functions of deep learning algorithms are almost all non-convex, and it is more difficult to converge after adding noise to the objective function. Moreover, simply adding noise to the well-trained model parameters may remarkably distort the model utility.

V. Discussion

Even though the techniques mentioned above ensure that the results can satisfy the requirement of differential privacy and prevent models from revealing inappropriate details of sensitive data, there remain great research challenges and also open several avenues for further investigations.

Model utility. The prevalent mechanisms such as the Laplace or Gaussian mechanism, can be introduced to numerous machine learning methods by virtue of their flexibility and simplicity. With a lot of noise being added, the foremost challenge for researchers is ensuring the accuracy of analysis results, especially for those models that require high accuracy. As a different scheme, the output or objective perturbation mechanism involves differential privacy into various learning algorithms to avoid the privacy disclosure of samples. In ERM-based techniques, random noise is added to the output models or the convex objective functions of models. Meanwhile, ERM has essential constraints that the objective function must be convex and *L*-Lipschitz. The PAC-based techniques try to measure the relationship between the quantity of learning samples and the bounded accuracy of the model. Obviously, the accuracy of model results is in direct proportion to the number of learning samples. By doing so, the privacy learning by PAC may lead to higher sample complexity even gain inaccurate results or become impractical for real applications. However, the mentioned perturbation mechanisms in this survey may be not the only ways to develop a differentially private model. For addressing potential privacy issues in machine learning, how to design an effective differentially private algorithm still requires further exploration.

Privacy level. To provide the principled and rigorous privacy guarantees in fundamental machine learning models, the privacy budget should be bounded at first. Then the composition bounds can be used to ensure that the final model has desirable differentially private properties and retains acceptable model utility. However, the sensitivity of the output with respect to changing a single input record is usually hard to determine especially for various specific problems. Consequently, different ways should be tried to reduce the privacy budget of differential privacy and tighten the utility bound of machine learning methods. Moreover, there still lacks of a unified framework that satisfies differential privacy and can be applied to different machine learning approaches.

Applications. Another challenging problem is how to achieve privacy preserving in evolutionary computation and fuzzy systems, which are fascinating research areas in computational intelligence. For example, an important next step is to incorporate differential privacy mechanisms to operators, evaluation functions and solutions when designing data-driven evolutionary algorithms. Especially, adding appropriate noise may increase the diversity of the population. It would also be an exciting research area to explore adding calibrated noise to the process of fuzzy reasoning, such as affiliation functions.

Last but not least, incorporating differential privacy into complex models represents a long-standing challenge for research communities of privacy preserving. On one hand, complex models can lead to the increased complexity in managing the risk of information disclosure, which means that excessive sanitization is usually indispensable and even leading to distorting the data utility for analysis. On the other hand, preserving privacy in complex models may consume more time and even become computationally infeasible. Especially in deep learning, training complex neural networks may cost a considerable amount of time. For these reasons, it is nontrivial for researchers to apply differentially private techniques to different types of complex machine learning models for a wide variety of specific tasks. Many other techniques, such as designing novel distributed protocols, may provide future opportunities for improving utility and privacy of differentially private machine learning. Other machine learning models have the potential to be performed with differential privacy.

VI. Conclusion

Hitherto, most of the existing machine learning models are known to implicitly memorize many details of training datasets during training and inadvertently reveal privacy during model prediction. It is paramount to improve the non-private machine learning methods for non-experts on privacy especially for those who majored in information-critical domains. Throughout this paper, we give a comprehensive review of privacy preserving in machine learning under the unified framework of differential privacy. We provide an intuitive handle for the operator to gracefully balance between utility and privacy, through which more users can benefit from machine learning models built on their sensitive data. And finally, we discuss major challenges and promising research directions in the field of differentially private machine learning.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant no. 2017YFB0802200), the Fundamental Research Funds for Central Universities and the Innovation Fund of Xidian University.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2012, pp. 1097–1105. doi: 10.1145/3065386.

[2] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. ACM SIGSAC Conf. Computer and Communication Security*, Dallas, Oct. 2017, pp. 587–601. doi: 10.1145/3133956.3134077.

[3] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learning Representation*, San Juan, Puerto Rico, May 2016.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security and Privacy*, San Jose, CA, May 2017. doi: 10.1109/SP.2017.41.

[5] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Computer Communication Secur*ity, Denver, CO, Oct. 2015, pp. 1322–1333. doi: 10.1145/2810103.2813677.

[6] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. 25th Security Symp.*, Austin, TX, Aug. 2016, pp. 601–618. doi: 10.5555/3241094.3241142.

[7] 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 Security Symp.*, San Diego, CA, Aug. 2014, pp. 17–32.

[8] 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," *Int. J. Secur. Netw.*, vol. 10, no. 3, pp. 137–150, 2015. doi: 10.1504/IJSN.2015.071829.

[9] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, vol. 10, no. 05, pp. 557–570, July 2002. doi: 10.1142/S0218488502001648.

[10] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Engineering*, Apr. 2006, pp. 24–24. doi: 10.1109/ICDE.2006.1.

[11] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. 23rd Int. Conf. Data Engineering*, Istanbul, Turkey, Apr. 2007, pp. 106–115. doi: 10.1109/ICDE.2007.367856.

[12] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proc. Int. Conf. Knowledge Discovery Data Minining*, Las Vegas, NV, Aug. 2008, pp. 70–78. doi: 10.1145/1401890.1401904.

[13] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proc. 31st Int. Conf. Very Large Data Bases*, Aug. 2005, pp. 901–909.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Conf. Theory Cryptography*, New York, Mar. 2006, pp. 265–284. doi: 10.1007/11681878_14.

[15] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *Proc. 17th Symp. Principles Database Syst*ems, Seattle, WA, June 1998, p. 188. doi: 10.1145/275487.275508.

[16] M. E. Nergiz, M. Atzori, and C. Clifton, "Hiding the presence of individuals from shared database," in *Proc. ACM SIGMOD Int. Conf. Management Data*, Beijing, June 2007, pp. 665–676. doi: 10.1145/1247480.1247554.

[17] F. Mcsherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. IEEE Symp. Foundations Computer Sci*ence, Providence, RI, Nov. 2007, pp. 94–103. doi: 10.1109/FOCS.2007.66.

[18] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. Advances Neural Information Processing Systems*, Vancouver, B.C., Canada, Dec. 2008, pp. 289–296.

[19] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014. doi: 10.1561/0400000042.

[20] Z. Ji, Z. C. Lipton, and C. Elkan, Differential privacy and machine learning: A survey and review. 2014. [Online]. Available: arXiv:1412.7584

[21] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. 25th Annu. Int. Conf. Theory Applications Cryptographic Techniques*, Saint Petersburg, Russia, May 2006, pp. 486–503. doi: 10.1007/11761679_29.

[22] F. D. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," *Commun. ACM*, vol. 53, no. 9, pp. 89–97, Sept. 2010. doi: 10.1145/1810891.1810916.

[23] K. Nissim and S. Raskhodnikova, "Smooth sensitivity and sampling in private data analysis," in *Proc. 39th Annu. ACM Symp. Theory Computing*, San Diego, CA, June 2007, pp. 75–84. doi: 10.1145/1250790.1250803.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. doi: 10.1038/nature14539.

[25] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proc. Advances Neural Information Processing Systems*, June 2014, pp. 2924–2932. doi: 10.5555/2969033.2969153.

[26] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong, "Differentially private naive bayes classification," in *Proc. Int. Conf. Web Intelligence*, Atlanta, GA, Nov. 2013, pp. 571–576. doi: 10.1109/WI-IAT.2013.80.

[27] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private naive Bayes learning over multiple data sources," *Inf. Sci.*, vol. 444, pp. 89–104, Feb. 2018. doi: 10.1016/j.ins.2018.02.056.

[28] A. Blum, Dwork, F. Mcsherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *Proc. 24th ACM. Symp. Principles Database Systems*, Baltimore, MD, June 2005, pp. 128–138. doi: 10.1145/1065167.1065184.

[29] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proc. Int. Conf. Knowledge Discovery Data Mining*, Washington, D.C., July 2010, pp. 493–502. doi: 10.1145/1835804.1835868.

[30] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," *IEEE Trans. Data Privacy*, vol. 5, no. 1, pp. 273–295, Dec. 2009. doi: 10.1109/ICDMW.2009.93.

[31] S. Rana, S. K. Gupta, and S. Venkatesh, "Differentially private random forest with high utility," in *Proc. 15th Int. Conf. Data Min*ing, Atlantic City, NJ, Nov. 2015, pp. 955–960. doi: 10.1109/ICDM.2015.76.

[32] X. Bai, J. Yao, M. Yuan, K. Deng, X. Xie, and H. Guan, "Embedding differential privacy in decision tree algorithm with different depths," *China Sci.*, vol. 60, no. 8, pp. 1–15, Nov. 2016. doi: 10.1007/s11432-016-0442-1.

[33] S. Fletcher and M. Z. Islam, "Differentially private random decision forests using smooth sensitivity," *Expert Syst. Appl.*, vol. 78, pp. 16–31, July 2017. doi: 10.1016/j.eswa.2017.01.034.

[34] X. Liu, Q. Li, T. Li, and D. Chen, "Differentially private classification with decision tree ensemble," *Appl. Soft Comput.*, vol. 62, pp. 807–816, Jan. 2018. doi: 10.1016/j.asoc.2017.09.010.

[35] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996. doi: 10.1007/BF00058655.

[36] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. 51st Annu. IEEE Symp. Foundations Computer Science*, Las Vegas, NV, Dec. 2010, pp. 51–60. doi: 10.1109/FOCS.2010.12.

[37] F. Dan, A. Fiat, H. Kaplan, and K. Nissim, "Private coresets," in *Proc. 41th Annu. ACM Symp. Theory Comput*ing, Bethesda, MD, June 2009, pp. 361–370. doi: 10.1145/1536414.1536465.

[38] Y. Wang, Y. X. Wang, and A. Singh, "Differentially private subspace clustering," in *Proc. Advances Neural Information Processing Systems*, Montrèal, Dec. 2015, pp. 1000–1008.

[39] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin, "Differentially private k-means clustering," in *Proc. 6th Conf. Data Application Security and Privacy*, New Orleans, LA, Mar. 2016, pp. 26–37. doi: 10.1145/2857705.2857708.

[40] D. Su, J. Cao, N. Li, E. Bertino, M. Lyu, and H. Jin, "Differentially private k-means clustering and a hybrid approach to private optimization," *ACM Trans. Priv. Secur.*, vol. 20, no. 4, pp. 16:1–16:33, Oct. 2017. doi: 10.1145/3133201.

[41] V. Schellekens, A. Chatalic, F. Houssiau, Y.-A. de Montjoye, L. Jacques, and R. Gribonval, "Differentially private compressive k-means," in *Proc. 44th Int. Conf. Acoustics Speech Signal Processing*, Brighton, May 2019, pp. 7933–7937. doi: 10.1109/ICASSP.2019.8682829.

[42] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "Near-optimal algorithms for differentially-private principal components," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2905–2943, 2012.

[43] Y. Chikuse, *Statistics on Special Manifolds*, vol. 174. New York: Springer-Verlag, 2012.

[44] M. Kapralov and K. Talwar, "On differentially private low rank approximation," in *Proc. 24th Annu. Symp. Discrete Algorithms*, New Orleans, LA, Jan. 2013, pp. 1395–1414. doi: 10.1137/1.9781611973105.101.

[45] W. Jiang, C. Xie, and Z. Zhang, "Wishart mechanism for differentially private principal components analysis," in *Proc. AAAI Conf. Artificial Intelligence*, Austin, TX, Jan. 2015, pp. 1730–1736.

[46] A. Gonen and G. Ran-Bachrach, Smooth sensitivity based approach for differentially private principal component analysis. 2017. [Online]. Availabe: arXiv:1710.10556

[47] H. Imtiaz and A. D. Sarwate, Distributed differentially-private algorithms for matrix and tensor factorization. 2018. [Online]. Availabe: arXiv:1804.10299

[48] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *Proc. 25th Annu. Conf. Learning Theory*, Edinburgh, Scotland, June 2012, pp. 24.1–24.34.

[49] B. Kulis and P. L. Bartlett, "Implicit online learning," in *Proc. 27th Int. Conf. Machine Learning*, Haifa, Israel, June 2010, pp. 575–582.

[50] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Machine Learning*, Washington, D.C., Aug. 2003, pp. 928–936.

[51] N. Agarwal and K. Singh, "The rice of differential privacy for online learning," in *Proc. 34th Int. Conf. Machine Learning*, Sydney, Australia, Aug. 2017, pp. 32–40.

[52] C. Li, P. Zhou, L. Xiong, Q. Wang, and T. Wang, "Differentially private distributed online learning," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1440–1453, Jan. 2018. doi: 10.1109/TKDE.2018.2794384.

[53] L. Jing, "Differentially private M-estimators," in *Proc. Advances in Neural Information Processing Systems*, Granada, Spain, Dec. 2011, pp. 361–369.

[54] Z. Zhang, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," in *Proc. 31st Int. Conf. Very Large Data Bases*, July 2012, vol. 5, no. 11, pp. 1364–1375.

[55] K. Ligett, S. Neel, A. Roth, B. Waggoner, and S. Z. Wu, "Accuracy first: Selecting a differential privacy level for accuracy constrained ERM," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, Jan. 2017, pp. 2563–2573.

[56] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res*, vol. 12, pp. 1069–1109, 2009.

[57] H. Li, L. Xiong, Z. Ji, and X. Jiang, "Partitioning-based mechanisms under personalized differential privacy," in *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining*, Apr. 2017, pp. 615–627. doi: 10.1007/978-3-319-57454-7_48.

[58] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, B.C., Canada, Dec. 2008, pp. 1177–1184.

[59] B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, Learning in a large function space: Privacy-preserving mechanisms for SVM learning. 2009. [Online]. Available: arXiv:0911.5708

[60] P. Jain and A. Thakurta, "Differentially private learning with kernels," in *Proc. 30th Int. Conf. Machine Learning*, Atlanta, June 2013, pp. 118–126.

[61] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distributed Computing and Networking*, Goa, India, Jan. 2015, pp. 4:1–4:10. doi: 10.1145/2684464.2684480.

[62] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 395–408, Mar. 2018. doi: 10.1109/TCNS.2016.2614100.

[63] D. Angluin, "Computational learning theory: Survey and selected bibliography," in *Proc. 24th Annu. Symp. Theory of Computing*, B.C., Canada, May 1992, pp. 351–369. doi: 10.1145/129712.129746.

[64] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *Proc. ACM Symp. Theory of Computing*, Victoria, B.C., Canada, May 2008, pp. 609–618. doi: 10.1145/1374376.1374464.

[65] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM J. Comput.*, vol. 40, no. 3, pp. 793–826, June 2011. doi: 10.1137/090756090.

[66] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," *Mach. Learn.*, vol. 94, pp. 401–437, Sept. 2013. doi: 10.1007/s10994-013-5404-1.

[67] A. Beimel, K. Nissim, and U. Stemmer, "Characterizing the sample complexity of private learners," in *Proc. 4th Innovations in Theoretical Computer Science*, Berkeley, Jan. 2013, pp. 97–110. doi: 10.1145/2422436.2422450.

[68] K. Chaudhuri and D. Hsu, "Sample complexity bounds for differentially private learning," in *Proc. 24th Annu. Conf. Learning Theory*, Budapest, Hungary, July 2011, pp. 155–186.

[69] A. Beimel, K. Nissim, and U. Stemmer, "Learning privately with labeled and unlabeled examples," in *Proc. 26th Annu. Symp. Discrete Algorithms*, San Diego, CA, Jan. 2015, pp. 461–477. doi: 10.1137/1.9781611973730.32.

[70] B. Amos, K. Nissim, and U. Stemmer, "Private learning and sanitization: Pure vs. approximate differential privacy," in *Proc. Int. Workshop on Randomization and Approximation Techniques in Computer Science*, Berkeley, Aug. 2013, pp. 363–378. doi: 10.1007/978-3-642-40328-6_26.

[71] M. Bun, K. Nissim, U. Stemmer, and S. Vadhan, "Differentially private release and learning of threshold functions," in *Proc. IEEE 56th Symp. Foundations of Computer Science*, Berkeley, Oct. 2015, pp. 634–649. doi: 10.1109/FOCS.2015.45.

[72] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal and Information Processing*, Austin, TX, Dec. 2013, pp. 245–248. doi: 10.1109/GlobalSIP.2013.6736861.

[73] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Computer and Communications Security*, Vienna, Austria, Oct. 2016, pp. 308–318. doi: 10.1145/2976749.2978318.

[74] M. Chase, R. Gilad-Bachrach, K. Laine, K. Lauter, and P. Rindal, "Private collaborative neural network learning," Tech. Rep., 2017. [Online]. Available: https://eprint.iacr.org/2017/762

[75] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, Learning differentially private language models without losing accuracy. 2017. [Online]. Available: arXiv:1710.06963

[76] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, Dec. 2017, pp. 715–720.

[77] P. Lu and Yu, "Poster: A unified framework of differentially private synthetic data release with generative adversarial network," in *Proc. ACM SIGSAC Conf. Computer and Communications Security*, Dallas, Nov. 2017, pp. 2547–2549. doi: 10.1145/3133956.3138823.

[78] B. K. Beaulieu-Jones, Z. S. Wu, C. J. Williams, and C. S. Greene, Privacy-preserving generative deep neural networks support clinical data sharing. 2017. [Online]. Available: bioRxiv:159756

[79] X. Zhang, S. Ji, and T. Wang, Differentially private releasing via deep generative model. 2018. [Online]. Available: arXiv:1801.01594

[80] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, Differentially private generative adversarial network. 2018. [Online]. Available: arXiv: 1802.06739

[81] Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. ACM SIGSAC Conf. Computer and Communications Security*, Denver, CO, Oct. 2015, pp. 909–910. doi: 10.1145/2810103.2813687.

[82] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive Laplace mechanism: Differential privacy preservation in deep learning," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, Nov. 2017, pp. 385–394. doi: 10.1109/ICDM.2017.48.

[83] X. Zhang, S. Ji, H. Wang, and T. Wang, "Private, yet practical, multiparty deep learning," in *Proc. 37th Int. Conf. Distributed Computing Systems*, Atlanta, GA, June 2017, pp. 1442–1452. doi: 10.1109/ICDCS.2017.215.

[84] N. Papernot, M. Abadi, U. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *Proc. Int. Conf. Learning Representations*, San Juan, Puerto Rico, May 2017.

[85] S. Das, S. Datta, and B. B. Chaudhuri, "Handling data irregularities in classification: Foundations, trends, and future challenges," *Pattern Recog.*, vol. 81, pp. 674–693, Sept. 2018. doi: 10.1016/j.patcog.2018.03.008.

[86] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson, "Scalable private learning with PATE," in *Proc. Int. Conf. Learning Representations*, Vancouver, B.C., Canada, May 2018.

[87] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems*, June 2014, pp. 2672–2680.

[88] J. Jordon, J. Yoon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *Proc. Int. Conf. Learning Representations*, New Orleans, LA, May 2019.

[89] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren, "GANobfuscator: Mitigating information leakage under GAN via differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2372–2386, Feb. 2019. doi: 10.1109/TIFS.2019.2897874.

[90] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn*, vol. 2, no. 1, pp. 1–127, 2007. doi: 10.1561/2200000006.

[91] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proc. AAAI Conf. Artificial Intelligence*, Phoenix, AZ, Feb. 2016, pp. 1309–1316.

[92] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Int. Conf. Machine Learning*, New York, June 2009, pp. 609–616. doi: 10.1145/1553374.1553453.

[93] N. Phan, X. Wu, and D. Dou, "Preserving differential privacy in convolutional deep belief networks," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1681–1704, July 2017. doi: 10.1007/s10994-017-5656-2.

[94] D. Xu, S. Yuan, X. Wu, and H. Phan, "DPNE: Differentially private network embedding," in *Proc. Asia Pacific Conf. Knowledge Discovery and Data Mining*, June 2018, pp. 235–246. doi: 10.1007/978-3-319-93037-4_19.

[95] B. Perozzi, Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. Int. Conf. Knowledge Discovery Data Mining*, New York, Aug. 2014, pp. 701–710. doi: 10.1145/2623330.2623732.

[96] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and nod2vec," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Marina Del Rey, CA, Feb. 2018, pp. 459–467. doi: 10.1145/3159652.3159706.