IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2022, accepted February 9, 2022, date of publication February 15, 2022, date of current version March 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151670

Differential Privacy for Deep and Federated Learning: A Survey

AHMED EL OUADRHIRI AND AHMED ABDELHADI, (Senior Member, IEEE)
Department of Engineering Technology, University of Houston, Houston, TX 77204, USA
Corresponding author: Ahmed El Ouadrhiri (aeouadrh@central.uh.edu)

ABSTRACT Users' privacy is vulnerable at all stages of the deep learning process. Sensitive information of users may be disclosed during data collection, during training, or even after releasing the trained learning model. Differential privacy (DP) is one of the main approaches proven to ensure strong privacy protection in data analysis. DP protects the users' privacy by adding noise to the original dataset or the learning parameters. Thus, an attacker could not retrieve the sensitive information of an individual involved in the training dataset. In this survey paper, we analyze and present the main ideas based on DP to guarantee users' privacy in deep and federated learning. In addition, we illustrate all types of probability distributions that satisfy the DP mechanism, with their properties and use cases. Furthermore, we bridge the gap in the literature by providing a comprehensive overview of the different variants of DP, highlighting their advantages and limitations. Our study reveals the gap between theory and application, accuracy, and robustness of DP. Finally, we provide several open problems and future research directions.

INDEX TERMS Deep learning, federated learning, privacy protection, differential privacy, probability distribution.

I. INTRODUCTION

In recent years, deep learning (DL) demonstrates a big success in many fields such as Healthcare, Marketing, Transportation, etc. For example, DL is used for early disease detection [1]–[3], predicting the future and adapting to the market needs [4], [5], helping people with disabilities [6], facilitating our daily activities [7]. To produce models with high accuracy, DL requires big datasets for training the model. However, datasets may contain sensitive information [8] that should not be disclosed to any third party, which raises concerns about the privacy protection in DL. In fact, users' privacy is threatened even when attackers do not have direct access to the dataset. Attackers may query the trained learning model to recover the original training dataset [9]– this type of attack is called model inversion attacks. There is another type of attack called membership inference attacks [10] where attackers' aim is to distinguish whether an individual was part of the training dataset or not. We refer the reader to [11] for a well-presented review of research work for different privacy attack types facing DL. Therefore, ensuring users' privacy in DL is of great importance.

The associate editor coordinating the review of this manuscript and approving it for publication was Yan Huo.

In this paper, we present the different techniques proposed to tackle the privacy issues in deep and federated learning (FL). Particularly, we focus on differential privacy (DP) which became a *de facto* standard for protecting users' privacy in statistical computations. These techniques can be divided into three categories:

- Techniques protecting users' privacy before publishing a dataset such as k-anonymity, l-diversity, and t-closeness. These techniques produce a new dataset, called a privacy-preserving (PP) dataset, protecting users' sensitive information. Attackers could not learn any critical information even if they have full access to the dataset.
- Techniques protecting users' privacy during the training. These techniques allow collaboratively training a model between many clients (i.e., parties) while keeping the dataset of each client private.
- DP-based techniques. DP may protect users' privacy in the three stages of training a DL model namely: 1) Before the training by producing PP datasets. 2) During the training by protecting the gradients sent from clients to the server in the case of collaborative training. 3) After the training by producing DL models resistant to model inference and model inversion attacks. DP is also used for protecting users' privacy while interrogating a database. This is because an attacker with some

background knowledge can perform some count and sum queries on a database and hence conclude the sensitive information of the victim.

In the rest of this section, we provide an overview of the approaches proposed in each category. Then, we present the different review works done on privacy protection in DL. Afterward, we outline the contributions of this paper compared to the recent literature in the field of DP applied to DL.

1) TECHNIQUES PRODUCING PRIVACY-PRESERVING DATASETS

$k-$**anonymity** [12], [13] is a mechanism for ensuring privacy before releasing a dataset. $k-$anonymity consists of generalizing quasi-identifier attributes and redacting some others so a record cannot be distinguished from the least $k-1$ other records in the dataset, in other words, the probability of re-identification is $\frac{1}{k}$. Nevertheless, $k-$anonymization performs poorly on the anonymization of a high-dimensional dataset and does not provide strong protection against attribute disclosure [14], [15]. An attacker with some background knowledge of victims could infer critical information about them.

$l-$**diversity** [16] has been proposed to overcome the $k-$anonymity shortcomings. It is based on the $k-$anonymity principle, i.e., generalizing quasi-identifier attributes and redacting some others so we cannot distinguish a tuple from at least $k-1$ other tuples. Then divides the dataset into q-block, where each block contains k tuples with the same values of the quasi-identifier attributes. In addition, $l-$diversity ensures that each block has l distinct values for the sensitive attribute. Hence, $l-$diversity provides strong privacy against background knowledge and homogeneity attacks. The larger the value of l is, the stronger the privacy is guaranteed.

$t-$**closeness** [17] which covers some drawbacks of $l-$diversity especially when the values of the sensitive attributes could take only two values (i.e., when $l=2$). The $t-$closeness mechanism is also based on the $k-$anonymity principle to create $t-$closeness classes (blocks) for sensitive attributes. A class is said to have $t-$closeness for a sensitive attribute A if the earth mover distance [18]) between the distribution of A in the class and in the dataset is not higher than a threshold t. A dataset is said to have $t-$closeness if all classes satisfy $t-$closeness. By limiting the distance between classes and the whole dataset, the amount of useful information that an adversary can learn from the quasi-identifier values of an individual and the distribution of the class is limited and does not reveal precious information. Since it limits disclosure about the correlation between quasi-identifier attributes and the sensitive attribute.

2) TECHNIQUES PROTECTING USERS' PRIVACY DURING THE TRAINING

Secure multiparty computing (SMC) [19], [20] is a subfield of cryptography that allows creating methods to jointly compute a function using inputs from different parties without revealing those inputs neither to each other nor to the central server. Thus, SMC does not require a trusted third party. Figure 1 illustrates an example of calculating the average salary of 3 clients without revealing the salary neither to the central server nor to other clients. In the first step, each client splits his/her salary into three pieces. In step 2, each client keeps one piece and shares the remaining two pieces with other clients. For example in Figure 1, client 1 divides his/her salary to $-30, 50, 60$, keeps -30 locally, and shares 50 with the second client and 60 with the third client. In step 3, each client aggregates the received pieces with his/her local piece and calculates the sum. In our example, the first client aggregates the pieces receive from the second and third client to calculate the sum as $60 = 50 + 40 + (-30)$. In step 4, clients send the calculated sum to the central server. Finally, in step 5, the server calculates the average of the received values, which is in this example $\frac{60+130+90}{3} = 93.33$. Thus, We get the same average as if we calculated the average using the true values of the salary. An SMC protocol is said to be secure if it satisfies the following properties:

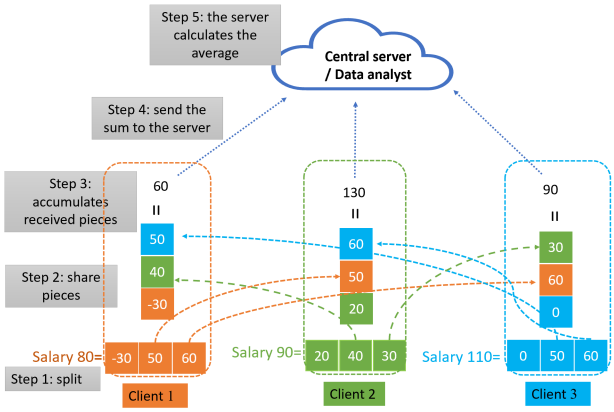

FIGURE 1. Secure multiparty computing.

- Privacy: A client should not be able to learn any information about any other client in the network, except the information that can be derived from his/her own input and output.
- Correctness: The output received by each participant should be correct.
- Independence of input: The inputs of malicious clients must be independent of the inputs of the honest clients.
- Guarantee of output: Malicious clients should not be able to prevent legitimate clients from receiving their outputs.
- Fairness: Malicious clients receive their outputs if and only if honest participants receive their outputs.

There are some recent works using SMC in federated learning (FL) to protect the privacy of clients [21]–[23]. However, SMC is costly in terms of computational complexity and communication overhead. Thus, SMC is unsuitable for training complex models over big datasets implicating many clients.

Homomorphic encryption (HE) [24] provides strong privacy protection as it allows training a model on an encrypted dataset. HE achieves the same accuracy as if the training was performed on the unencrypted version (i.e., original dataset) of the dataset [25], [26]. However, using HE in DL is inefficient in practice due to its computational complexity, especially when the training dataset is too large to fit in the computer memory. HE is more suited for MLaaS [27], [28] when the model is already trained and ready to use. In this case, users send their input encrypted to the cloud that makes the prediction. Then the cloud sends back the results encrypted to the users.

3) DIFFERENTIAL PRIVACY

Recently, **DP** [29], [30] has attracted a great deal of attention in DL, especially in guaranteeing users' privacy. DP allows analyzing a dataset without revealing a single individual private information. In other words, analyzing the dataset and computing statistics about it (such as mode, median, mean, etc.) does not allow revealing the information that an individual's information was included in the original dataset or not.

Although the first definition of DP back in 2006, it does not receive attention in practical use only in the last few years. The main reason that may prevent using DP in practice is the accuracy. In fact, the accuracy decreases by increasing the level of privacy protection. To overcome this problem, researchers either try to find a trade-off between accuracy and privacy [31], [32] or combine DP with another technique (e.g., memorization, adding a proxy server) for strengthening the privacy protection [33], [34]. There are many applications of DP in practice. For example, Google proposed RAPPORT [35], an approach based on DP for privately collecting statistics from devices of clients (e.g., Software hangs and time of utilization). Microsoft [33] applies DP with the memorization technique for privately collecting statistics periodically from their clients' devices. Apple [36] also used DP to collect statistics from their clients' devices to enhance their quality of experience. DP is also adopted by the US census bureau to protect the publications of the 2018 End-to-End Census Test. DP applications can be divided into two categories:

1) Central differential privacy (CDP), as defined in [29], requires that users trust the database holder (i.e., the data curator) to keep their privacy. CDP consists of adding random noise after collecting the data from individuals. The random noise is added to the original dataset or to the results of queries launched on the original dataset.
2) Local differential privacy (LDP) [37], [38] overcomes CDP shortcomings and ensures privacy when individuals do not trust the data curator. During data collection, individuals perturb and/or encode their responses before submitting them to the central server. LDP mechanisms should be carefully implemented, as each individual perturbs his response individually, the estimated frequencies on the dataset may not be inaccurate [39].

All DP schemes have the same principle which is adding noise to protect the sensitive information of individuals. Certainly, adding more noise guarantees perfect protection of privacy. On the other hand, adding less noise allows attackers to reveal sensitive information about individuals. Recently, Ren *et al.* [9] succeeded to recover the original dataset when a small noise is added to the gradient. Thus, based on what we will discuss in section II, one has to evaluate the privacy leakage for a given privacy budget ϵ before publishing a dataset, a learning model, or responding to a query function.

A. RELATED WORKS

Fatemehsadat *et al.* [40] present a summary of information disclosure attacks to better situate the need for privacy protection in DL. The authors divide PP methods into three categories: 1) methods for PP datasets that protect the privacy of clients in a dataset, 2) methods protecting the privacy of clients during the training phase, 3) methods for PP models that protect the privacy of clients after deploying the trained model. However, the authors do not detail DP and PP methods for FL models, they only provide a brief introduction to FL and split learning (SL) without detailing the state-of-art PP methods proposed in FL. Ha *et al.* [41] detail the inference attacks and present methods for producing PP DL models. They categorize these methods into three groups: 1) gradient-level methods that consist of adding noise to the gradient, 2) function-level methods that consist of adding noise to the loss function, and 3) label-level methods that consist of adding noise to the label set during the training. Amine *et al.* [42] provide a review of 45 papers handling the problem of PP in DL. The authors present different works that are based on different techniques such as DP and HE on top of the strongest approaches, in addition to model splitting [43], mimic learning [44], and partial parameters sharing [45]. All presented works are dated before July 2019, nevertheless, the period after 2019 till now, had recognized the emergence of many works especially for preserving privacy in FL. Xue *et al.* [46] provide a detailed explanation of the different attacks that may threaten a DL model. More specifically, they categorize these attacks into five types: 1) data poisoning attacks, 2) backdoor attacks, 3) adversarial examples attacks, 4) model stealing attacks, 5) recovery of sensitive training data which includes model inversion attacks and membership inference attacks. Accordingly, the authors present some approaches to deal with the different attacks, including approaches for producing PP models. Chang *et al.* [47] present a summary of privacy issues in DL. The authors divide these problems into two types: issues during training and issues during prediction, i.e., after deploying the trained model. Accordingly, The authors present some countermeasures approaches to deal with these issues. Zhang *et al.* [48] discuss PP approaches proposed to deal with attacks threatening collaborative learning. The authors categorize these

approaches into two categories: 1) PP during the training phase, and 2) PP after deploying the trained model.

All survey works on privacy in DL focus on detailing possible attacks against DL and presenting the different PP methods to protect the users' privacy. Yet, we didn't find any paper detailing DP in DL as well as presenting the different variants of DP proposed so far. The main differences between the contributions of the present survey and the above state-of-the-art works are summarized in Table 1.

B. CONTRIBUTIONS

This paper presents a detailed survey of DP mechanisms designed for PP in DL and FL, we bridge the gap of the existing literature by providing:

- A comprehensive description of the probability distributions that satisfy the ϵ-DP definition with their use cases.
- A detailed description of ϵ-DP variants, namely (ϵ, δ)-DP, (α, ϵ)-rényi DP, and f-DP, comparing the privacy leakage due to composition.
- A review of the different works based on DP for protecting users' privacy in DL and FL. We divide these approaches into three categories based on their type of application: 1) PP queries, 2) PP datasets, 3) PP models.
- An analysis of the main ideas and recent approaches based on DP regarding the computational complexity, communication cost, and accuracy. This analysis illustrates the gap between theory, application, accuracy, and robustness of DP and brings forth many future research directions.

II. DIFFERENTIAL PRIVACY AND ITS VARIANTS

A. ϵ-DP

The main objective of DP is to allow studying the properties of a dataset (about a population) as a whole without revealing one's individual information. In other words, DP consists of adding noise to either statistical queries or the original dataset so that an adversary cannot know whether a particular individual is included in the dataset or not. DP as defined first in [29] requires that users trust the data curator since users send their correct data without any modification to the data curator. The data is stored in the central server as received from users. Nevertheless, the data curator does not trust the third party or the data analyst. Hence, the data curator uses DP to perturb the original dataset before responding to statistical queries of third parties for analysis. This type of implementation of DP is called Central DP. The name central DP comes from the fact that the perturbation is done centrally at the data curator, unlike the local DP which will be discussed in the next section.

We refer to a mechanism that satisfies DP by ϵ-DP [29], where ϵ denotes the privacy loss or privacy leakage. Before providing the definition of an ϵ-DP mechanism, we define the meaning of two neighboring datasets and the sensitivity of a given query function f.

Definition 1 (Neighboring datasets): Let D^n be the domain of all datasets, also known as dataset universe. $D, D' \in D^n$ are called neighboring if D and D' differ in one entry, i.e. one entry is added or removed from D to get D'.

Definition 2 (Sensitivity of a query function f): Given two neighboring datasets D and D', and a query function $f : D^n \rightarrow \mathbb{R}^d$ mapping databases to real numbers. The sensitivity of the function f is defined as the maximum value by which f changes if a single individual is added or removed from a dataset, this is formulated as

$$\Delta_f = \max_{D,D'} ||f(D) - f(D')||_1. \tag{1}$$

where $||\cdot||$ denotes the ℓ_1 norm. [1]

Definition 3 (ϵ-differential privacy): A mechanism or an algorithm M is called ϵ-differentially private if for all neighboring datasets $D, D' \in D^n$, and for all $S \subseteq Y$, where Y is the set of all possible outputs, we have:

$$Pr[M(D) \in S] \leq e^{\epsilon} Pr[M(D') \in S] \tag{2}$$

that is to say the output when the mechanism M is applied to D is similar to the output when M is applied to D'. The smallest ϵ is the perfect the privacy is guaranteed

It is worth mentioning that the combination, known in the literature by composition, of two DP mechanisms is also a DP mechanism (see proof in [49]). The composition theorem is defined as follows.

Theorem 1 (Composition): Let M_1 is an ϵ_1-DP mechanism and M_2 is an ϵ_2-DP mechanism. Then, the composition of M_1 and M_2 defined by $M_{1,2} = (M_1, M_2)$ is an $(\epsilon_1 + \epsilon_2)$-DP.

The composition theorem allows using DP in practical use cases such as guaranteeing the privacy of gradient in FL. For example, if a client in FL applies an $\epsilon-$DP mechanism to the gradient before sending it to the central server. After k epochs, the $\epsilon-$DP mechanism results in (due to the composition theorem) $(k \times \epsilon)-$DP mechanism. That is to say, the privacy leakage at the first epoch was ϵ, and after k epochs, the privacy leakage becomes $k \times \epsilon$.

B. (ϵ, δ)-DP

The first definition of ϵ-DP was introduced by Dwork *et al.* [29] as stated earlier in Definition 3. Afterward, the same authors proposed another relaxation of ϵ-DP called (ϵ, δ)-DP [50], [51] by adding δ as an additive term to the original definition. δ was added to capture the privacy protection of the Gaussian distribution (see Definition 6), as detailed in the previous subsection.

Definition 4 ((ϵ, δ)-DP [50], [51]): A mechanism M is called (ϵ, δ)-differentially private if for all neighboring datasets $D, D' \in D^n$. We have, for all $S \subseteq Y$, where Y is the set of all possible outputs:

$$Pr[M(D) \in S] \leq e^{\epsilon} Pr[M(D') \in S] + \delta \tag{3}$$

The interpretation of a mechanism M satisfies (ϵ, δ)-DP is this mechanism is ϵ-DP except with probability δ. That is to say, the mechanism M is ϵ-DP with probability $1 - \delta$.

[1] The ℓ_1 norm of a vector v, denoted by $||v||$, is the sum of the absolute values of the vector v.

TABLE 1. Comparison Between the Present Survey and the Existing State-of-the-art.

Ref.	Year	DP probability distributions	DP variants	PP datasets	PP query results	PP DL models	PP FL models
Ours	2021	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
[46]	2020	No	No	No	No	**Yes**	**Yes**
[40]	2020	No	No	**Yes**	No	**Yes**	No
[42]	2019	No	No	No	No	**Yes**	**Yes**
[41]	2019	No	No	No	No	**Yes**	No
[47]	2018	No	No	No	No	**Yes**	**Yes**
[48]	2018	No	No	No	No	No	**Yes**

(ϵ, δ)-DP is proposed to mitigate the privacy leakage of ϵ-DP under composition, as ϵ-DP is closed under composition [52]. (ϵ, δ)-DP provides smaller cumulative loss under composition. [2] (ϵ, δ)-DP is not appropriate in the scenario where S is a singleton set. It is worth mentioning that δ should be negligible compared to the size of the set S (i.e. $\delta \ll 1/|S|$), to avoid the worst-case scenario of always violating the privacy of a δ fraction of the dataset.

C. PROBABILITY DISTRIBUTIONS SATISFYING DP (ϵ-DP OR (ϵ, δ)-DP)

In this subsection, we present the different probability distributions proposed in the literature and satisfy either ϵ-DP or (ϵ, δ)-DP. We point out the type of noise generated, the condition of applications, and the use cases as well.

1) **Laplace mechanism [29]:** is the most used approach in literature as it can be used for any type of data [53]. The Laplace mechanism consists of adding a noise drawn from the continuous Laplace distribution $Lap(0, \frac{\Delta_f}{\epsilon})$.
Definition 5: Given a function $f : D^n \to Y$, where Y is the set of all possible outputs, and $\epsilon > 0$. The Laplace mechanism is defined as

$$M(D) = f(D) + Lap(0, \frac{\Delta_f}{\epsilon}). \tag{4}$$

2) **Gaussian mechanism [54]**: satisfies the principle of the new variant of ϵ-DP which is f-DP (see subsection II-E for more details), and support tractability of the privacy budget under composition.
Definition 6: Given two neighboring datasets D and D' in the dataset universe D^n, a query function $f : D^n \to Y$, where Y is the set of all possible outputs, and $\epsilon > 0$. The ϵ-Gaussian DP (ϵ-GDP) mechanism is defined as:

$$M(D) = f(D) + \mathcal{N}(0, \frac{\Delta_f^2}{\epsilon^2}), \tag{5}$$

where $\mathcal{N}(0, \frac{\Delta_f^2}{\epsilon^2})$ stands for the normal distribution.

[2] Composition means the sequential application of DP. For example, if we apply DP on the result of a query function f, thus calling the query function f one time is ϵ-DP, and calling the query function f sequentially k times is at least $(k \times \epsilon)$-DP.

3) **Geometric mechanism [55]:** used to add discrete noise to the result of a query function for integer-valued data type [56].
Definition 7: Given a dataset D, a query function $f : D^n \to Y$, and $\epsilon > 0$. The two-sided geometric mechanism adds independent noise to the query function f:

$$M(D) = f(D) + \Delta, \tag{6}$$

where Δ is a random variable with a two-sided geometric distribution:

$$P(\Delta = \delta) = \frac{1 - e^{-\epsilon}}{1 + e^{-\epsilon}} e^{-\epsilon|\delta|}$$

for every integer δ.
The probability $P(\Delta = \delta)$ can be interpreted as the probability of adding discrete noise δ to the result of the query function f. The Geometric mechanism is a discretized version of the Laplace mechanism [57].

4) **Exponential mechanism [58]:** is most suited when we have to select a noisy (i.e., random) response from the set of all possible outputs, instead of adding noise to the result of the query function [53].
Definition 8: Given a dataset universe D^n, a set of all possible outputs Y, and a scoring function $u : D^n \times Y \to \mathbb{R}$ which defines a score for each element $D \in D^n$ to each element $y \in Y$. That is to say, u assigns a real valued score to any pair (D, y) from $D^n \times Y$ with the understanding that higher scores correspond to most suited outputs.
The exponential mechanism consists of selecting an output $y \in Y$ with probability proportional to $e^{(\frac{\epsilon u(D,y)}{2\Delta u})}$. This means that the exponential mechanism returns an element from Y that has the highest score with probability $e^{(\frac{\epsilon u(D,y)}{2\Delta u})}$. Hence, the exponential mechanism sometimes returns $y \in Y$ which does not have the highest score.
The Laplace mechanism can be captured from the exponential mechanism by taking $u(D, y) = -|f(D) - y|$, where f is the function defined in Definition 5. $u(D, y)$ takes the maximal value when the query function result $f(D)$ is equal to the exact output value y.

5) **Binomial mechanism [59]:** used to add discrete noise to the result of the query function. However, the

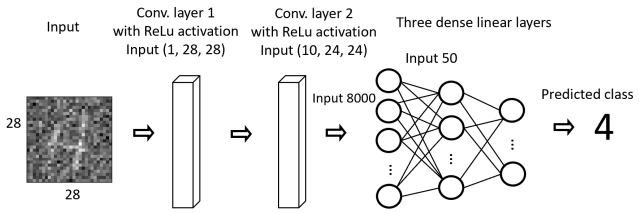

FIGURE 2. The learning model architecture.

Binomial mechanism satisfies (ϵ, δ)-DP ((ϵ, δ)-DP is a variant of ϵ-DP as defined in the next subsection) under constraints [59] as illustrated in the following definition.

Definition 9: Given two neighboring datasets D and D' in the dataset universe D^n, and a query function $f : D^n \rightarrow Y$, the Binomial mechanism is defined as:

$$M(D) = f(D) + (Z - Np)s \tag{7}$$

where $Z \sim Bin(N, p)$, and $s = \frac{1}{t}$ is the quantization scale for some $t \in \mathbb{N}$. s helps to normalize the noise correctly. The parameters δ, N, p, and s should satisfy the following condition:

$$Np(1-p) \geq max(23\log(10d/\delta), 2\Delta_\infty/s), \tag{8}$$

where d is the dimension of the output of the query function f, and Δ_∞ is the infinity norm of the sensitivity of the query function f.

D. ϵ-DP VERSUS (ϵ, δ)-DP

In this subsection, we compare the most used DP distributions in the literature, namely the Laplace (i.e., ϵ-DP) and the Gaussian distribution (i.e., (ϵ, δ)-DP), in terms of privacy protection and accuracy. For this purpose, we develop and train a learning model using three different scenarios according to the dataset used in the training:

- Scenario 1: In the first scenario, we train the learning model on the original MNIST dataset [60] without any noise. This is our reference scenario to evaluate the impact of privacy protection (i.e., noise) on accuracy.
- Scenario 2: In the second scenario, we train the learning model on the privacy-preserving MNIST dataset generated by adding Laplace noise.
- Scenario 3: In the third scenario, we train the learning model on the privacy-preserving MNIST dataset generated by adding Gaussian noise.

The learning model, see Figure 2, is composed of two convolutional layers. Each layer is associated with ReLu as an activation function. The second convolutional layer is associated with Dropout Regularization to prevent overfitting. Then, we add three fully connected linear layers with the dimension of the output of the last linear layer is 10, which corresponds to the number of classes that we have in our training dataset.

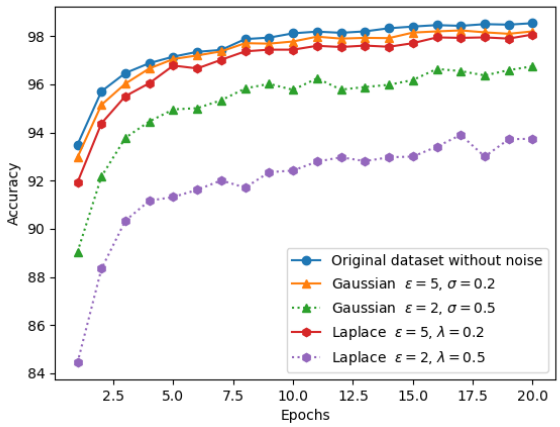

FIGURE 3. The accuracy of the learning model during the training for the three scenarios varing the privacy leakage ϵ.

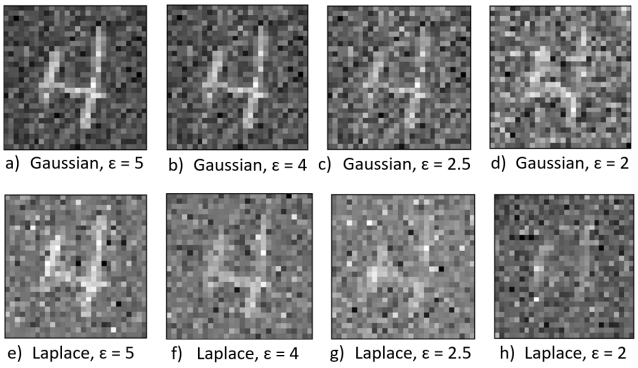

FIGURE 4. Sample of the digit 4 from the original and the privacy-preserving datasets with different privacy leakage ϵ.

Figure 3 illustrates the accuracy of the learning model for the three scenarios during 20 epochs of training. We notice that the highest accuracy is achieved for the first scenario (i.e., the blue curve), compared to the other two scenarios (scenarios 2, and 3). On the other hand, for scenarios 2 and 3 where we train the model on privacy-preserving datasets, we notice that the Gaussian distribution gives higher accuracy compared to the Laplace distribution. This endorses the theoretical analysis; As the Laplace distribution is ϵ-DP, and the Gaussian distribution is (ϵ)-DP with probability δ (i.e., (ϵ, δ)-DP).

This difference is illustrated in Figure 4, which shows samples from the privacy-preserving MNIST datasets generated using the Laplace and the Gaussian distributions. Images of the Laplace distribution are noisy compared to the Gaussian distribution, especially, when the privacy leakage ϵ decreases. For example, in the case of $\epsilon = 2$, we can still notice some white pixels for the Gaussian distribution (see subfigure 4-d). On the contrary, for the Laplace distribution (see subfigure 4-h), the image is totally noisy to the extent that we can't extract any useful information. Thus, the Laplace distribution guarantees strong privacy protection compared to the Gaussian distribution, but at the expense of accuracy.

TABLE 2. Summary of Probability Distributions Satisfying ϵ-DP/(ϵ, δ)-DP With Their use Cases.

Probability distribution	Noise	Use cases	Privacy leakage
Laplace [29]	Adds real values drawn from $Lap(0, \frac{\Delta_f}{\epsilon})$.	Protects queries' results, datasets, gradients.	ϵ-DP
Gaussian [54]	Adds real values drawn from $\mathcal{N}(0, \frac{\Delta_f^2}{\epsilon^2})$.	Protects queries' results, datasets, gradients.	(ϵ, δ)-DP
Geometric [55]	Adds a discrete value δ with probability $P(\Delta = \delta) = \frac{1-e^{-\epsilon}}{1+e^{-\epsilon}} e^{-\epsilon\lvert\delta\rvert}$.	Protects queries' results, datasets.	ϵ-DP
Exponential [58]	Chooses a random output with probability $e^{(\frac{\epsilon u(D,y)}{2\Delta u})}$.	Protects learning models.	ϵ-DP
Binomial [59]	Adds discrete value drawn from $(Bin(N, p) - Np)s$.	Protects queries' results, datasets.	(ϵ, δ)-DP

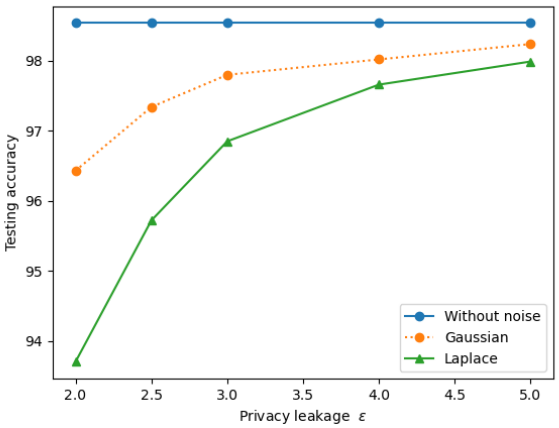

FIGURE 5. Impact of the privacy leakage ϵ on the accuracy.

Figure 5 illustrates the impact of privacy leakage ϵ on accuracy. In this figure, we plot the accuracy of the learning model using the testing dataset. The model is trained on different versions of the MNIST dataset, varying the privacy leakage. Overall, the accuracy decreases by decreasing the privacy leakage for the two distributions. In addition, the Gaussian distribution gives higher accuracy compared to the Laplace distribution; the difference increases by decreasing the privacy leakage. For example, in the case of $\epsilon = 5$, the Gaussian distribution gives an accuracy of 98.24 compared to 97.99 for the Laplace distribution. Whilst, in the case of $\epsilon = 2$, the Gaussian distribution gives an accuracy of 96.43 compared to 93.70 for the Laplace distribution.

E. VARIANTS OF DIFFERENTIAL PRIVACY

In this subsection, we present the pertinent variants of DP, namely (α, ϵ)-Rényi DP ((α, ϵ)-RDP), and f-DP. We state the main differences between these new definitions of privacy protection, as well as the advantages and disadvantages of each variant.

The most challenging problem of the DP mechanism is that the privacy leakage increases due to composition (see Theorem 1). In fact, the privacy leakage increases by increasing k, the number of compositions. Thus, determining a tighter bound of the privacy leakage due to composition allows learning more features (e.g., producing accurate learning models) from a dataset while protecting individuals' sensitive information.

Dwork *et al.* [61] determine a bound of the privacy budget after k composition defined by $(\sqrt{2\,k\,ln(\frac{1}{\delta'})} \times \epsilon + k \times \epsilon(e^{\epsilon} - 1), k\delta + \delta')$-DP for any $\epsilon, \delta, \delta' \in]0, \infty[$. Afterward, Kairouz *et al.* [62] define a procedure that allows achieving the optimal bound of the privacy budget after k queries. The authors prove that for any $i \in \{0, 1, \ldots, \lfloor k/2 \rfloor\}$, the composition of k queries satisfies

$$((k-2i)\epsilon, 1 - (1-\delta)^k(1-\delta_i)) - DP, \tag{9}$$

where $\delta_i = \frac{\sum_{\ell=0}^{i-1} \binom{k}{\ell}(e^{(k-\ell)\epsilon} - e^{(k-2i+\ell)\epsilon})}{(1+e^{\epsilon})^k}$.

Thus, in practice, we may use Eq. (9) to determine the optimal values of ϵ_i and δ_i for each query q to do not exceed a predefined privacy leakage (ϵ, δ) after k queries. Thereafter, Mironov [52] determine a tighter bound of privacy leakage due to composition using Rényi divergence-based DP, which is the subject of the next subsection.

1) (α, ϵ)−RÉNYI DIFFERENTIAL PRIVACY ((α, ϵ)−RDP)

Although the original definition of ϵ-DP provides strong privacy protection of data privacy, it still does not tightly handle the privacy leakage due to composition. The problem of composition appears also while training a federated learning model, as the privacy leakage increases by increasing the number of training epochs. For example, if we apply a mechanism M with a privacy loss ϵ at each epoch, consequently at the end of the training, we will result in a privacy loss of $k\epsilon$, where k is the total number of epochs during the training. This problem is of great importance, as the privacy leakage increases by increasing the number of training epochs.

Mironov [52] introduces a new relaxation of ϵ-DP based on the concept of Rényi divergence. This new variant of ϵ-DP allows accurate tracking of the privacy leakage due to composition. $(\alpha, \epsilon)-$RDP is defined as a generalization of the notion of differential privacy based on the concept of Rényi divergence [63]. $(\alpha, \epsilon)-$RDP provides a quantitatively accurate way of tracking cumulative privacy leakage under composition. Before defining (α, ϵ)-RDP, we will define the Rényi divergence [63]:

Definition 10 (Rényi divergence): Given two probability distributions P and Q defined over R, the Rényi divergence of order $\alpha > 1$ is

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log E_{x\sim Q}\left(\frac{P(x)}{Q(x)}\right)^\alpha, \tag{10}$$

with $P(x)$ is the density of P at x. The logarithm here is natural, and $x \sim Q$ means that x follows the distribution Q.

Definition 11 ((α, ϵ)-RDP): A mechanism $M : D^n \to Y$ is said to satisfy (α, ϵ)-RDP of order α, if for any neighboring datasets D, D', and for all $S \subseteq Y$, it holds that

$$D_\alpha(M(D)||M(D')) \leq \epsilon. \tag{11}$$

We also have the following inequality holds for (α, ϵ)-RDP:

$$Pr[M(D) \in S] \leq (e^\epsilon Pr[M(D') \in S])^{\frac{\alpha-1}{\alpha}}. \tag{12}$$

(α, ϵ)-RDP allows achieving a tighter bound, of privacy leakage due to composition, compared to the bound determined by [61], [62]. Using the (α, ϵ)-RDP definition, Mironov demonstrates the following corollary.

Corollary 1: Let $0 < \delta < 1$ such that $\log(1/\delta) \geq \epsilon^2 k$. The composition of k queries, each satisfies ϵ-DP, is (ϵ', δ)-DP where $\epsilon' = 4\epsilon\sqrt{2k\log(1/\delta)}$.

Thus, we may use this result to track the privacy leakage due to composition. For example, in federated learning, we may use this corollary to determine the scale of the Laplace distribution (i.e., $\lambda = 1/\epsilon_i$, where ϵ_i is the privacy leakage of each training epoch i calculated from corollary 1) in order to do not exceed a predefined privacy leakage (ϵ, δ).

The strong property of RDP is that the optimal privacy bound of k RDP is easily calculated using the addition, i.e., the composition of k, (α, ϵ)-RDP mechanism is $(\alpha, k\epsilon)$-RDP [52], [64]. Therefore, we can use RDP to calculate an optimal bound of the privacy leakage due to composition, and then convert the resulting RDP to (ϵ, δ)-DP using the following proposition 1:

Proposition 1: Given a mechanism M satisfies (α, ϵ)-RDP, then it also satisfies $(\epsilon(\alpha) + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-DP where $0 < \delta < 1$ and $\epsilon(\alpha) = \max\{D_\alpha(M(D)||M(D'))\}$.

Since this holds for all $\alpha > 1$, thus, the optimal privacy bound ϵ' can be determined by optimizing over α the following expression:

$$\epsilon' = \inf_{\alpha>1}\{k \times \epsilon(\alpha) + \frac{\log(1/\delta)}{\alpha - 1}\}. \tag{13}$$

Developing this expression for the case when M is a Gaussian mechanism, we get:

$$\epsilon' = \inf_{\alpha>1}\{k \times \frac{\alpha}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1}\}, \tag{14}$$

where σ^2 is the variance of the Gaussian distribution.

For the case when M is a Laplace mechanism, we get:

$$\epsilon' = \inf_{\alpha>1}\{k\frac{1}{\alpha-1}\log\{\frac{\alpha}{2\alpha-1}e^{(\frac{\alpha-1}{\lambda})} + \frac{\alpha-1}{2\alpha-1}e^{(\frac{-\alpha}{\lambda})} + \frac{\log(1/\delta)}{\alpha-1}\}, \tag{15}$$

where λ is the scale of the Laplace distribution.

Ultimately, the RDP allows determining a tighter bound, of the privacy leakage due to composition, compared to the start-of-the-art privacy bounds calculated using the original definition of (ϵ, δ)-DP [49], [61], [62].

2) f-DP

Dong *et al.* [54] propose f-DP, a new relaxation of ϵ-DP based on hypothesis testing interpretation. f-DP is parameterized by a function rather than parameters (e.g., ϵ, δ), which offers a complete characterization of privacy.

In fact, f-DP is based on the following simple idea of ϵ-DP: By interrogating two neighboring datasets D and D', an attacker can not conclude if an individual belongs to D or to D'. Thus, this problem can be formulated using the following two hypothesis testing:

- H_0: the underlying dataset is D,
- H_1: the underlying dataset is D'.

with the objective of making these two hypotheses indistinguishable. This is equivalent to find the optimal trade-off between the achievable type I error[3] and type II error. [4] More precisely, consider a rejection rule $\phi \in [0, 1]$, the type I error rate and the type II error rate are respectively defined as follows:

$$\alpha_\phi = \mathbb{E}_{M(D)}[\phi], \;\; \beta_\phi = 1 - \mathbb{E}_{M(D')}[\phi], \tag{16}$$

where $M(D)$ and $M(D')$ are the probability distributions of the mechanism M applied to the two datasets D and D', respectively. The two error rates satisfy the constraint of the total variation distance:

$$\alpha_\phi + \beta_\phi \geq 1 - TV(M(D), M(D')), \tag{17}$$

where the total variance distance $TV(M(D), M(D'))$ is the largest possible difference between the probabilities that the two probability distributions $M(D)$ and $M(D')$ can assign to the same event.

Therefore, the f-DP main objective is to characterize the fine-grained trade-off between type I and type II errors. That is to say, fixing type I error at any level and finding the

[3]Type I error, also known as false positive, represents the rejection of a null hypothesis H_0 given that it is true.

[4]Type II error, also known as false negative, represents the false alarm or the non-rejection of a null hypothesis H_0 given that it is false.

minimal achievable type II error. Before defining the f-DP, we define the trade-off function.

Definition 12 (Trade-off function): For any two probability distributions P and Q on the same space, the trade-off function $T(P, Q) : [0, 1] \rightarrow [0, 1]$ is

$$T(P, Q)(\alpha) = inf\{\beta_\phi : \alpha_\phi \leq \alpha\}, \tag{18}$$

where the infimum is taken over all (measurable) rejection rules.

The similarity of P and Q increases by increasing the value of the function $T(P, Q)(\alpha)$.

In practice, it is difficult to satisfy this definition. Thus, the following proposition presents a necessary and sufficient condition to determine a trade-off function.

Proposition 2: A function $f : [0, 1] \rightarrow [0, 1]$ is a trade-off function if and only if f is convex, continuous, non-increasing, and $f(x) \leq 1 - x$ for $x \in [0, 1]$.

f-DP is built on top of trade-off functions:

Definition 13 (f-DP): Let f be a trade-off function. A mechanism M is said to be f-DP if

$$T(M(D), M(D')) \geq f, \tag{19}$$

for all neighboring datasets D and D'.

This definition is explained as follows. Given two distributions P and Q such that $f = (P, G)$, a mechanism M satisfies f-DP means that distinguishing $M(D)$ and $M(D')$ is at least as difficult as distinguishing P and Q.

The Gaussian probability distribution function is an example of the functions that satisfy the f-DP Definition, where f is the trade-off function of two normal distributions. To be more specific, let $\epsilon > 0$, and

$$G_\epsilon := T\left(\mathcal{N}(0, 1), \mathcal{N}(\mu, 1)\right). \tag{20}$$

An explicit expression of the trade-off function G_ϵ is:

$$G_\mu(\alpha) = \Phi(\Phi^{-1}(1 - \alpha) - \mu), \tag{21}$$

where Φ is the Gaussian standard cumulative distribution function, and Φ^{-1} is its inverse function. Hence, the GDP is defined as follows:

Definition 14: Given two neighboring datasets D and D' in the dataset universe D^n. A mechanism M satisfies the μ-Gaussian DP (μ-GDP) if it is $G_\mu - DP$, i.e.,

$$T(M(S), M(S')) \geq G_\mu$$

This definition gives a necessary and sufficient condition for a mechanism M to satisfy μ-GDP. The Gaussian distribution satisfies μ-GDP and it is the tightest possible privacy bound of the Gaussian mechanism, see Section II-E, Definition 6.

Using this new definition based on a function (i.e., f-DP) allows determining a tighter bound of the privacy leakage due to composition. For the case of the Gaussian mechanism, the authors in [54] proved that:

Corollary 2: The composition of k queries, each satisfies μ-GDP, is $(\sqrt{k}\,\mu)$-GDP.

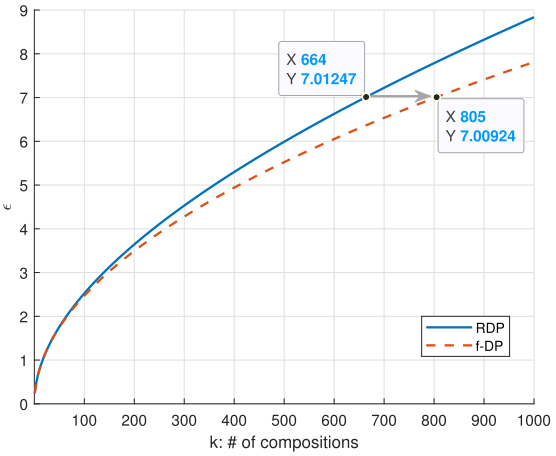

FIGURE 6. The privacy leakage ϵ'_G of M Gaussian mechanism, and the privacy leakage ϵ' of M RDP. We assume $\sigma = 20$ and $\delta = 10^{-5}$.

In [52], the authors prove that RDP guarantees a tighter bound compared to [64]. In the rest of this subsection, we compare the bound of RDP and f-DP. For that purpose, we have first to respond to the following question: What is the privacy leakage ϵ for a mechanism M satisfying μ-GDP.

Starting from [50], we conclude:

$$\sigma^2 = \frac{2\,\log(2/\delta)}{\epsilon^2}. \tag{22}$$

Thus, we can calculate the privacy leakage ϵ for a mechanism M satisfying μ-GDP using the following expression:

$$\epsilon = \sqrt{2\,\log(2/\delta)\,\mu^2}, \tag{23}$$

From this equation, we can calculate the privacy leakage ϵ' due to the composition of M mechanism each one of them satisfies μ-GDP:

$$\epsilon'_G = \sqrt{2\,\log(2/\delta)\,k\,\mu^2}, \tag{24}$$

In Figure 6, we compare the privacy leakage ϵ' (i.e., Eq. (14)) of M RDP mechanism and the privacy leakage ϵ'_G (i.e., Eq. (24)) of M GDP mechanism. We use the same parameters as [52], i.e., $\sigma = 20$, $\delta = 10^{-5}$. According to this figure, f-DP allows us to achieve smaller privacy bound by up to 1.025 of difference. This result has an important impact on private DL algorithms especially FL, as it allows for more training epochs for the same privacy budget ϵ, e.g., 141 more training epochs for any ϵ larger than 7.

Thus, in practice (e.g., training an FL model), given a predefined privacy budget ϵ' to do not exceed, we may use Eq. (24) to determine the optimal privacy budget ϵ of each training epoch. Then, using Eq. (22) we calculate the variance σ^2 of the Gaussian distribution from which we generate the noise to add.

In summary, RDP and f-DP are two new different definitions of DP, where RDP is based on Rényi divergence (parameterized by (α, ϵ)) and f-DP is based on hypotheses testing (parameterized by a trade-off function f). Comparing RDP

with f-DP in terms of privacy leakage due to composition, f-DP allows determining a tighter bound compared to RDP. Thus, in FL and for the same privacy budget ϵ, f-DP allows for more training epochs compared to RDP. We refer the reader to [65] for more details about the relationship between RDP, f-DP, and (ϵ, δ)-DP.

III. CENTRAL DIFFERENTIAL PRIVACY FOR DEEP LEARNING

In this section, we present the recent research works based on CDP for protecting the users' privacy in DL. We divide these works into three categories:

A. **PP learning models.** The main idea of these approaches is to add static or dynamic noise to the coefficients of the objective function.
B. **PP query results.** These approaches can be divided into two types: *i*) The works that add noise to the query result after running the query on the original dataset. *ii*) The works that partition the dataset, run the query on each part of the dataset, and then add noise to sub-queries results.
C. **PP datasets.** These approaches add noise to the original dataset for producing a new PP dataset.

Table 3 summarizes the presented works in this section. This table illustrates the main idea and the final objective of the contribution, along with the type of probability distribution used in the DP mechanism.

A. PRIVACY-PRESERVING LEARNING MODEL

DL models are threatened by inversion attacks [81], [82]. An attacker can reveal some sensitive information about an individual by interrogating the learning model and using background information about this individual. Information disclosure is done by linking the target features with the model outcomes. In the rest of this subsection, we present the recent research work that handles the problem of privacy in DL. These works apply DP during the training to produce PP models.

Pan *et al.* [66] present adaptive differentially private regression (ADPR) mechanism, a dynamic privacy noise allocation mechanism that takes into account the relevance of the input attributes to the outputs. The mechanism consists of adding Laplace noise drawn from $Lap(\frac{\Delta_f}{\epsilon_j})$ into the polynomial coefficients of the objective function. Δ_f is the sensitivity, and ϵ_j (the amount of privacy) is calculated according to the input's features relevance $R_j(D)$. Thus, less noise is added to attributes that highly impact the learning model and vice versa. Although this approach gives better accuracy compared to [67]–[69], [83], it is costly in terms of computation as it has to run a pre-processing learning step to determine the relevance of each attribute. The approaches of [66] and [67] are the same, except that [67] adds noise with the same privacy budget which may decrease the model's accuracy.

Fang *et al.* [69] decompose the objective function into monomial terms and add noise to each monomial term according to its sensitivity $\Delta(f_i)$ and the privacy budget ϵ_i. The privacy budget ϵ_i is dynamic and updated at each iteration and should satisfy $\epsilon_1 + \ldots + \epsilon_d = \epsilon$ where ϵ is the total privacy budget and d stands for the number of terms of the polynomial objective function.

Katrina *et al.* [70] propose a noise reduction framework for learning models based on empirical risk minimization (ERM) algorithms as a loss function. The framework consists of applying a privacy budget depending on a predefined accuracy. The framework adds noise to the model parameter (i.e., gradient) to generate a sequence of parameters, where each parameter corresponds to a privacy budget. Afterward, the framework selects the privacy budget that gives an accuracy higher than the given predefined threshold. This approach is costly in terms of computational complexity because it has to sequentially go over all noisy optimal parameters until finding the privacy budget ϵ that gives the predefined accuracy.

Ultimately, as illustrated by our simulation results (see Figure 5), the accuracy decreases by decreasing the privacy leakage ϵ (i.e., introducing more noise). That is to say, the accuracy decreases by increasing the privacy protection. Thus, determining the amount of privacy leakage ϵ that guarantees both a perfect privacy protection and acceptable accuracy is challenging and depends on the application scenario. There are three categories of works in the literature: The first category consists of predefining an acceptable accuracy c and then determines the optimal privacy leakage ϵ that guarantees the highest privacy protection and an accuracy greater than the predefined accuracy c [53], [70]. The second category consists of predefining the privacy leakage that should be guaranteed and then determining the learning model parameters the maximize the accuracy [84]. The third category consists of adding noise based on the relevance of each input feature to the outputs [66], [67].

B. PRIVACY-PRESERVING QUERY RESULTS

Privacy leakage may occur even if an adversary does not have direct access to the dataset but he/she can perform some count or summation queries on the dataset [85], [86]. Figure 7 illustrates how DP is used at the data curator to protect the clients' privacy. Specifically, the process involves three main steps. In Step 1, users send their personal information to the data curator. In step 2, users' data is protected and aggregated in a database. In Step 3, data analysts interact with the database via queries or request the whole dataset for a training purpose. Before responding to the data analyst queries, the data curator guarantees the users' privacy by adding random noise either to the query results or to the values of the attributes of the dataset.

Earlier, Hay *et al.* [75] propose to add Laplace noise $Lap(\frac{1}{\epsilon})$ to the set of results of queries q, and send the noisy results $\tilde{q}$ to the data analyst. The resulting outputs are evaluated according to a set of constraints to guarantee the consistency of the results. When the noisy results are inconsistent, a post-processing step called constrained inference is added to calculate $\overline{q}$, the new consistent results of the

TABLE 3. Summary of Contributions in Central Differential Privacy.

Ref.	Year	DP-mechanism	Main contribution	Objective
[66]	2021	Laplace	Adding dynamic noise to the polynomial coefficients of the objective function during the training process.	PP learning model.
[67]	2012	Laplace	Adding static noise to the objective function during the training process.	PP learning model.
[68]	2019	Laplace	Adding dynamic noise according to the relevance of the coefficients in the objective function.	PP learning model.
[69]	2019	Laplace	Adding dynamic noise to the objective function according to the relevance of the coefficient and ensuring that the cumulative noise added during the training does not exceed ϵ.	PP learning model.
[70]	2017	Laplace	Adding dynamic noise according to a predefined accuracy level.	PP learning model.
[71]	2014	Laplace	Construct a noisy frequent pattern tree by adding half of the noise to the support σ and adding the second half of the noise after calculating the count queries.	PP frequent itemsets.
[53]	2020	Exponential	Handle the problem of heterogeneous data in one attribute and add noise according to the predefined level of accuracy.	New anonymized dataset for classification purposes.
[72]	2019	Laplace	Normalize the dataset and add noise according to the relevance of the attribute to guarantee a predefined level of privacy.	PP normalized dataset.
[56]	2021	Geometric	Construct a region hierarchy τ that corresponds to the dataset D and add noise subject to constraints.	PP dataset.
[73]	2021	Laplace	Constructing a latent tree T of the dataset based on the correlations between the attributes.	Generating a new synthetic dataset.
[74]	2021	Laplace	Using n-gram model for sequential datasets and adding noise according to the level of the leaf in the model.	Generating a PP sequential dataset.
[75]	2010	Laplace	Adding Laplace noise to count query results while ensuring consistency of the new noisy results.	PP count queries.
[76], [77], [78]	2010, 2010, 2020	Laplace	Partition the dataset into partitions, calculate the count query on each partition, and add noise to each query before aggregating all the queries' results into one response.	PP count queries.
[79]	2014	Laplace and Gaussian	Propose a framework that consists of adding noise (with correction) to real-time series data received from end-users and estimating the time series data when the time series data are not received from end-users.	PP time series data.
[80]	2020	Laplace noise	Classify the social network into groups within a graph and then apply DP to protect the nodes, edges, degrees, and structure of the graph.	PP social network dataset.

results of queries. $\overline{q}$ is the minimum l_2-norm solution which is the closest set to $\tilde{q}$ that satisfies the predefined constraints. Xiao *et al.* [76] propose two algorithms: 1) Cell-based algorithm, and 2) K-d tree-based Algorithm for partitioning the dataset into partitions and then adding Laplace noise to the result of the query on each partition before aggregating these results and responding to the main query. Cormode *et al.* [87] generalize the approach proposed by [75], [76] and propose to add non-uniform noise to the results of queries. That is to say, instead of adding noise with the same privacy budget ϵ to each partition, the authors propose to define a specific privacy budget for each partition. The privacy budget of each partition

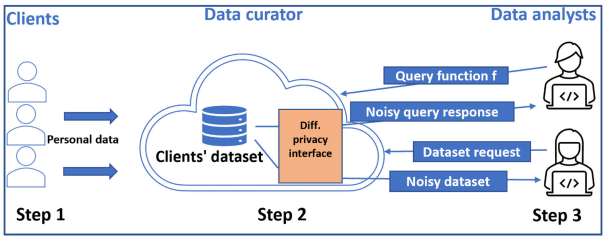

FIGURE 7. Central differential privacy for releasing privacy-preserving (noisy) datasets/queries.

is determined by minimizing the error $Err(Q)$, where $Err(Q)$ is equal to the variance of the noisy results of the queries.

There are many other works [77], [78], [88]–[90] that handle the problem of privacy for responding to a batch of queries. They use ϵ-DP to introduce Laplace noise and perturb the query results for protecting the user's privacy. For example, Huang *et al.* [78] decompose the original query set Q into orthogonal query subsets to construct another query set $\tilde{Q}$, such that each query from Q can be represented by elementary queries $\tilde{q}_i$ from $\tilde{Q}$. The Laplace noise is added to the query set $\tilde{Q}$ instead of Q which reduces the noise variance and then leads to better efficiency. Li *et al.* [77] represent the main query as a set of linear base queries on the dataset's attributes. These linear queries are represented as a matrix where each row contains the coefficients of the linear query. Then, the authors apply DP on the matrix to get PP result of the main query.

C. PRIVACY-PRESERVING DATASETS

Nowadays, many organizations release their clients' datasets to third parties for training DL models that help in making decisions [91]–[93]. However, providing dataset to a third party may violate users' privacy and breach privacy laws [94], [95]. Therefore, there is a great need for mechanisms that allows releasing datasets for analysis without revealing users' privacy or any sensitive information. DP proved to provide strong privacy protection while allowing datasets analysis. This is a hot research topic where research work can be divided into two categories: 1) The first category is the works producing pre-processing datasets balancing between accuracy and privacy for a specific learning model, such as frequent itemset mining models [71], [96], [97], and classification and clustering models [53], [72], [98], [99]. 2) The second category is the works producing a PP dataset of the original dataset [56], [73], [100], [101]. Next, we will present the relevant research work based on DP to protect users' privacy before publishing a dataset.

Wang *et al.* [53] propose a differentially private approach for heterogeneous dataset [5] for cluster analysis. The original dataset D is pre-processed using a clustering algorithm to get an initial cluster structure D^*. Then, the authors apply DP to the new dataset D^* to get the anonymized dataset D'. Sun *et al.* [72] normalize the dataset rows which makes the dataset distribution more concentrated. Afterward, the authors use classification and regression tree (CART) [102] to apply DP in function of the relevance/impact of each attribute on the classification results.

Lee et el. [71] propose an approach to release noisy dataset for frequent itemset learning; first, the algorithm takes an integer k and distinguish the top k most frequent items by running frequent itemset mining algorithm [103], after that, the algorithm builds an ϵ-differentially private FP-tree [104] that is released to the analyzer. The privacy allocation is based on two phases: 1) perturbing the threshold τ of the support itemset σ_k to be $\hat{\tau} = \sigma_k + Lap(.)$, and 2) adding a Laplace noise to the originally calculated support $\sigma(X)$.

Fioretto *et al.* [56] handle the problem of releasing a dataset of a large population without leaking sensitive information about individuals. The original dataset is restructured into a tree T of levels and groups (e.g., level 1 may design the country and level 2 may design the state and so on, and a group may be the households owning three cars), the DP consists of adding noise that should satisfy three conditions. *i*) **Consistency**: The sum of the groups' sizes of a specific level r and group s after adding the noise should be equal to the sum before. *ii*) **Validity**: The size of a specific level r and a group s after adding the noise should be non-negative integers. *iii*) **Faithfulness**: The group sizes at each level l of the hierarchy should be equal to the total count of groups G. The problem is solved using three approaches: 1) Direct optimization-based mechanism, 2) dynamic programming 3) polynomial-time mechanism by exploiting the structure of the cost tables. Tang *et al.* [73] present a stronger privacy protection approach called differentially private latent tree (DPLT). It consists of generating a new synthetic dataset from vertically partitioned data (i.e., the dataset is shared between many data curators where each one holds some attributes of the dataset. Data curators share a common identifier attribute). The approach is based on the latent tree model (LTM) [100] and contains three main steps: 1) generating latent attributes by condensing original attributes and adding Laplace noise to guarantee ϵ-DP, 2) quantifying the correlation (i.e., mutual information) between any two latent attributes, and finally, 3) constructing the latent tree T based on the previously calculated correlations. The authors assume that each data curator uses the same privacy budget ϵ, which may contribute to decreasing the accuracy and/or leaking data privacy. In fact, the privacy budget may depend on the attributes, hence determining the best privacy budget for each curator is still yet to explore. Mohammed *et al.* [101] handle the same problem, however, the proposed approach is heavy in terms of computation and communication and only applicable to two data curators.

There are many works that specifically handle the case of sequential datasets[6] [74], [105], [106]. The most relevant one is [74] which handles the problem of releasing and

[5] A heterogeneous dataset is a dataset composed of relational data and set-valued data, such as information of patients: gender (relational), age (relational), medical history (set-valued), etc.

[6] A sequential dataset contains records placed one after another, so, reading the dataset is done one by one starting from the first record.

guarantying privacy for sequential datasets. It produces a new sequential dataset $\tilde{D}$ based on n-gram model which provides a good trade-off between storage and accuracy; n-gram models are based on the property of Markov independence assumption to estimate the probability of the new node in the leaf. The privacy is guaranteed by adding adaptable (with regards to privacy budget and the length of a root-to-leaf path) Laplace noise to each node in the tree. Fan *et al.* [79] propose FAST, a framework to collect time-series statistics. First, the time-series data is sent from clients (i.e., end-users) to a trusted server and then the trusted server sends the collected data to third parties. The trusted server guarantees data privacy before sharing it with third parties. The FAST framework is based on a filtering component method that consists of adding Laplace noise to data points received from end-users (i.e., real-time data points received from end-users, these points are called sampling points), or predicting the data points in the case when the data points are not received from end-users (i.e., these points are called non-sampling points). This framework is not efficient when the response time is crucial. In addition, the privacy of users is not fully protected as the server access to users' information. Huang *et al.* [80] propose an approach to generate a PP dataset. The authors handle the special case of social networks where the dataset is represented by a graph $G(V, E)$. They apply a classification algorithm based on K-means clustering to classify the graph into T groups. Afterward, the authors apply four different privacy protection algorithms to protect nodes, edges, degrees, and structure of the graph: 1) The first algorithm protects the graph's structure by adding noise to the original graph after the clustering to get a new PP graph denoted by $G'(V', E')$. The resulting graph is different from the original one. 2) The second algorithm protects nodes by adding Laplace noise to each group. The noise is randomly added to the nodes. 3) The third algorithm disturbs the degree sequence to protect specific nodes from identification by an attacker. 4) The fourth algorithm is a post-processing step that consists of adding noise to nodes with a small degree to protect edges. This approach [80] provides a higher privacy protection level and data availability compared to the approaches proposed in [107] and [108], however, the authors did not evaluate the impact of the proposed approach on the accuracy.

IV. LOCAL DIFFERENTIAL PRIVACY FOR DEEP LEARNING

Local DP has been proposed to ensure users' privacy when individuals do not trust the central data curator (i.e., the cloud or the central server). The client adds noise or falsifies his answer with a probability before sending his response to the central data curator [109]. LDP was implicitly introduced in [38], [50] and first formalized in [37]. Although the idea of LDP is relatively old, it has only recently seen many real applications such as privately collecting data [33], [35], [36], and privately train FL models [110]. In the rest of this section, we present the relevant works and ideas based on LDP to protect users' privacy from inference and inversion attacks in FL. In addition, we will go over the three real applications

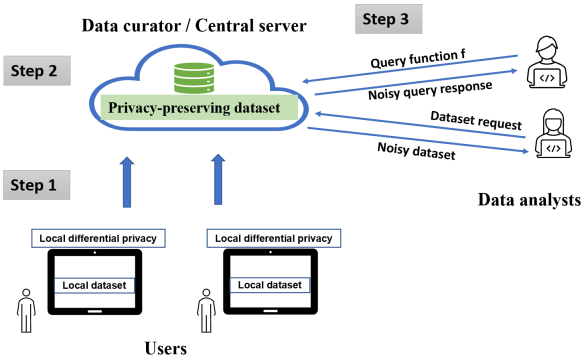

FIGURE 8. Local differential privacy.

implementing LDP for privately collecting data from end-users.

Figure 8 illustrates how LDP collects data from end-users while ensuring users' privacy. Specifically, the process involves three main steps. In step 1, users send their personal data after introducing noise to their responses. In step 2, the data curator collects the PP data from users and stores it in the database. In step 3, the data analysts could interrogate the database by launching direct queries or requesting the whole dataset. Unlike CDP where noise is added by the data curator, in LDP the data curator does not add any noise as the noise was added at users' level in step 1.

A. PRIVACY-PRESERVING FEDERATED LEARNING

FL [110] is a machine learning structure where many devices (e.g., mobile devices, laptops, organizations, etc.) collaboratively train a learning model under the orchestration of a central server. The main property of FL is that the dataset is not centralized, i.e., each device trains the model on its local dataset and then sends the updated parameters (e.g., gradient) to the central server. We refer the reader to [121] which is a good paper that presents in detail FL's characteristics, challenges, and research directions.

FL provides significant privacy improvement, as the local data of users is not explicitly sent to the central server. However, this is not enough for strong users' privacy protection. Since, a malicious user (or the server) could reconstruct the users' local dataset using only the local gradient sent to the server [10], [122], [123]. There are many recent research works tackling the problem of users' privacy in FL. Table 4 summarizes the presented works in this subsection, showing the key idea and the final objective of the contribution along with the type of probability distribution (i.e., type of noise). We divide these works into three main categories.

The first category is the works combining DP with another method/tool (e.g., Homomorphic encryption, secure multiparty computing, etc.) to protect the clients' privacy. Gong *et al.* [118] propose a framework for protecting the privacy of clients participating in an FL model based on DP and homomorphic encryption. DP is used to protect the gradients of clients from the central server by adding a noise drawn from a Laplace distribution. The amount of privacy

TABLE 4. Summary of Contributions Using Local Differential Privacy to Protect Users' Privacy in FL.

Ref.	Year	DP-mechanism	Main contribution	Objective
[111]	2021	Gaussian	Approach for efficient hierarchical caching in fog computing. The authors use FL for collaborative training and DP to protect the IoT devices' privacy.	Protecting the privacy of IoT devices while training an FL model for content popularity prediction.
[112]	2021	Gaussian	Functional encryption mechanism to secure the communication between the clients and the server. The privacy of clients is protected using DP.	PP model while ensuring the privacy of clients during the training process.
[34]	2021	Gaussian	A new PP approach that consists of only sharing partial parameters of the client's gradient with the server. The authors introduce a proxy server between the clients and the server to ensure the anonymity of the gradients.	PP model while ensuring the privacy of clients during the training process.
[113]	2021	Gaussian	Determining the standard deviation of the Gaussian distribution to achieve a predefined privacy leakage after T synchronization rounds. Proposing an algorithm for adjusting T to get the best convergence performance.	PP model while ensuring the privacy of clients during the training process.
[31]	2021	Gaussian	Decentralized learning by a token τ transiting in the network via peer-to-peer communication. The token is updated sequentially by each device before sending it to another device.	PP model while ensuring the privacy of clients during the training process.
[32]	2021	Gaussian	Characterize the Gaussian noise variance σ^2 required to guarantee a target privacy budget ϵ after T synchronization rounds.	Protecting the clients' privacy during the training and reducing the communication overhead.
[84]	2020	Laplace	Formalize an optimization problem subject to communication overheard, accuracy, and privacy budget.	Protecting the clients' privacy during the training and reducing the communication overhead.
[114]	2020	Gaussian	Asynchronous FL model with LDP to reduce the communication overhead and a mechanism to detect malicious nodes.	Protecting the clients' privacy during the training and reducing the communication overhead.
[115]	2020	Binomial	Formalize an optimization problem to determine the transmission rates allocation for the clients with regards to the privacy budget and the communication constraints.	Protecting the clients' privacy during the training and reducing the communication overhead.
[116]	2020	Gaussian	Apply DP in federated multi-task learning to protect the clients' privacy. The multi-task learning is used to deal with heterogeneous datasets.	Protecting the privacy of clients participating in federated multi-task learning.
[117]	2020	Randomized response	Approach to protect the privacy of clients while training a DL model with a cloud server. The first layers of the DL model are located at the clients with a new layer called LATENT to protect the privacy, and the last layers are located at the cloud server.	Protecting the clients' privacy while collaboratively training a DL model.
[118]	2020	Laplace	Framework to protect the privacy of clients participating in an FL model using DP and homomorphic encryption.	Protecting the privacy of clients while training an FL model.
[119]	2021	Uniform, Laplace, Gaussian	Framework called SDTF to protect the privacy of clients while training an FL model without a server. The privacy is protected using DP and Elgamal cryptosystem.	Protecting the privacy of clients while training a decentralized FL model.
[120]	2021	Random noise	Framework called chain-PPFL to protect the privacy of clients in an FL network using secure multiparty computing.	Protecting the privacy of clients while training a decentralized FL model.

budget ϵ increases dynamically with iterations (i.e., epochs), for example, in the first epoch the privacy budget is ϵ_{min}, then in the second epoch the privacy budget is increased by $\epsilon_{min} + c \times \frac{\epsilon_{max} - \epsilon_{min}}{\gamma}$ where c is the current epoch, and γ is the number of epochs to reach ϵ_{max}. Homomorphic encryption is used by each client to encrypt the gradient sent to the server. It is used to protect gradients of clients from a malicious server, in this case, all clients should share the same encryption key. Hence, even when the server colludes with a client and gets the key, it will not be able to retrieve the true values of clients' gradients as they are already protected using DP. However, recently, authors in [9] were able to conclude useful information about the original dataset even gradients were protected using DP.

Li *et al.* [120] propose a privacy-preserving FL framework based on secure multiparty computing called chain-PPFL. This approach is similar to [31], except that here the authors are based on the principle of secure multiparty computing instead of DP. First, the server sends the global model to all clients and initiates a token $\tau \in \mathbb{R}^d$ (where d is the dimension of gradient) with a random value. This token is sent to a client chosen from all clients. This latter updates the token by adding its gradient to it and sends the newly updated token to a randomly chosen client from its neighbors, and so on, until the last client sends the token to the server. The server subtracts the initially attributed value to the token and calculates the global weights of the next round. This process is repeated until the learning model converges. Using this approach, the privacy of clients is protected from their neighbors and also from the server. Since the server will receive the aggregated local gradient and will not be able to distinguish the local gradient of each client. Comparing chain-PPFL with other privacy-preserving approaches based on DP [34], [111], [112], chain-PPFL provides strong privacy (equivalent to an FL with $0-$DP) if clients do not collude with the server to attack a specific client which is not always guaranteed. In terms of accuracy, chain-PPFL provides higher accuracy as the noise added to the token, in the beginning, is subtracted at the end when the server receives the aggregated local gradients. The major issue of this approach is that the clients should trust each other and do not collude to attack one of them. In addition, using this decentralized strategy to aggregate local gradients make the FL network vulnerable to label-flipping and data poisoning attacks [124], [125], besides, it is difficult for the server to distinguish malicious from legitimate clients.

Wu *et al.* [116] propose to use DP for multi-task learning models. As the multi-task learning paradigm [126], [127] is to leverage useful knowledge in multiple tasks to improve the generalization performance of all tasks, the authors propose that each device in FL learns a task-specific parameter ω_i with the objective function f_i. The parameter ω_i is sent to the global server for learning the global task parameter ω_{M+1} which in turn is sent back to the clients and so on until the algorithm converges. For ensuring privacy, each client perturbs its own parameter according to Gaussian distribution $\mathcal{N}(0, \frac{\Delta_f^2}{\epsilon^2})$, where Δ_f^2 is the sensitivity of the average of the local gradients and ϵ is the privacy budget. The advantage of this approach is its ability to learn over multiple clients holding heterogeneous datasets, however, it is vulnerable to label-flipping attacks and also to model inversion attacks [9].

The second category is the works based on DP and the structure of the FL network (e.g., adding a proxy server, using a decentralized architecture) to protect the privacy of clients. Cyffers *et al.* [31] propose a new relaxation approach of LDP that allows analyzing data belonging to various devices while achieving a good trade-off between utility and privacy. FL is done by peer-to-peer communication from one node to another without a central server handling the communication. The proposed approach is a fully decentralized protocol where participants have only a local view of the studied system. The learning is made by a token τ transiting in the network. The token is updated sequentially by the device receiving it. Before realizing the token, each node adds random noise to the contribution to ensure differential privacy. This process is repeated K (a predefined value) times before getting the final model. The big issue of this contribution is its vulnerability to label-flipping and data poisoning attacks. An attacker could easily be infiltrated into the network and ruin the learning process.

Tran *et al.* [119] propose a framework, called Secure Decentralized Training Framework (SDTF), to protect the privacy of clients participating in training a decentralized FL. The clients train a model without a server, however, at each epoch, they elect a master node (one of them) which calculates the global gradient and sends it to all nodes, and so on until the algorithm converges. Each client perturbs his local gradient before sending it to the master node to protect his privacy. This framework achieves good accuracy since the master node, before sending the updated global gradient, estimates the sum of all noises added by clients and pulls it out from the global gradient. However, this framework cannot protect the clients' privacy from the master node and it cannot protect the privacy of a client if all other nodes collude against him/her.

Through several experiments, Zhao *et al.* [34] illustrate that sharing partial parameters of the gradient may almost achieve the accuracy of sharing all the parameters. Based on these results, the authors propose a PP learning approach which consists of sharing only some parameters of the local gradient and adding Gaussian noise to these parameters before sharing them with the server. The proposed approach is based on [64] to determine the noise amount to add and also to control the privacy leakage through synchronization rounds (i.e., composition). Besides, the authors propose to add a proxy between the clients and the server to ensure the anonymity of clients, therefore the server cannot distinguish from which client receives a certain gradient. The authors propose a strong method for protecting the clients' privacy, However, it would be of great importance to evaluate the

robustness of the proposed approach against inference and model inversion attacks [9].

Yin *et al.* [112] propose a PP approach that combines functional encryption and Bayesian differential privacy. The authors use functional encryption (FE) [128] to protect the communication between clients and the server. The FE proposed mechanism, called Multi-Input Function Encryption (MIFE), requires the help of a trusted third party (TTP) that provides a public key to clients and a private key to the server to encrypt/decrypt the gradient by the client/server. In addition, the authors are based on Bayesian DP [129] to 1) protect the privacy of clients by adding Gaussian noise to gradient depending on the data distribution, and 2) to track the privacy leakage due to synchronization rounds of FL. In order to reduce the communication cost between the clients and the server and also reduce computation cost at the server-side, the authors propose a method called sparse differential gradient where clients, at each synchronization round, send gradient to the server only if the gradient experienced a massive change (i.e., higher than a predefined threshold) compared with the previous gradient of the last synchronization round. Although this approach secures the communication and protects the gradient from the server, it requires a trusted third party that may launch model inversion attacks using the gradient received from clients.

The third category is the works whose objective is to protect privacy using DP and at the same time reduce the resource consumption, such as the energy and the communication overhead. Liu *et al.* [114] handle the problem of communication overhead and data privacy in federated edge learning for edge computing in the Industrial Internet of Things (IIoT). The authors propose:

1) an asynchronous model update to reduce the computation time that edge nodes wait for global model aggregation. Edge nodes send their gradients once they finish the local training without waiting for the next synchronization round by the server, this enhances the communication efficiency.
2) utilize LDP to mitigate gradient leakage attacks. The LDP mechanism is deployed at the edge nodes to protect the gradient.
3) a cloud-side malicious node detection method to detect malicious nodes. The detection of malicious nodes relies on the accuracy of the model parameters (e.g., gradient) sent by these nodes to the server. A node is characterized as malicious if the accuracy of its model parameters is lower than a dynamically calculated threshold ρ_{th}.

However, the proposed malicious node detection mechanism may discriminate some legitimate nodes from participating in the learning process. Indeed, the accuracy of these legitimate clients may be low than ρ_{th} in the first training rounds.

Sonee *et al.* [115] handle the problem of privacy and communication for training a federated stochastic gradient descent (SGD) model where the communication between the clients and the server takes place over a multiple access channel (MAC). The problem is formulated as an optimization problem aiming to determine the transmission rates allocation for the clients in the MAC to achieve the maximum convergence rate while satisfying the privacy and communication constraints.

Hu *et al.* [84] tackle the problem of resources constraints, accuracy, and privacy using FL in the Internet of Things (IoT). The authors' contribution is based on the assumption that each device should perform multiple local training epochs before sending the model updates to the server, instead of sending the model updates after each training epoch. This reduces the number of communication rounds and hence reduces the communication overhead. The problem has been formulated as an optimization problem to find the best model parameters (Θ^*) which guarantee the constraints on t and ϵ, where t is the number of epochs an IoT device should perform before sending the updates to the central server and ϵ is the minimum achievable privacy.

Mahawaga *et al.* [117] handle privacy in the case where multiple clients try to train a DL model using a convolutional neural network (CNN). TThe first convolutional layers of the model are placed at the clients' side along with a new layer called LATENT. This new layer is responsible for protecting the privacy of clients via LDP; The authors propose an approach called modified optimized unary encoding (MOUE) [130] that consists of randomizing the bit's vector 1 and 0 differently before sending the output to the server. The last layers are placed at the cloud server that communicates with the clients via software-defined networks (SDN) and network function virtualization (NFV). The simulation results show that the proposed approach achieves high accuracy (up to 90%) with a lightly high privacy budget (i.e., $\epsilon = 0.5$), however, the authors did not evaluate the privacy leakage due to composition.

Kim *et al.* [32] study the trade-off between the privacy budget, utility, and communication rate for an SGD FL model. The authors characterize the Gaussian noise variance σ^2 required to guarantee a target privacy budget after T rounds of weight updates between the clients and the server. The authors compare their works to [49], [61], [62], [64] and find that their approach requires the smallest noise variance for the same privacy budget ϵ.

Wei *et al.* [113] improve the work done by [64] for controlling the privacy leakage of DP through sequential composition and provide an explicit expression for calculating the standard deviation σ_i of the Gaussian distribution that should be used by a client i to guarantee a privacy leakage of $(\epsilon_i, \delta_i)-$LDP at the end of T synchronization round. The expression of σ_i of the client i depends on the sampling ratio q (i.e., q is the ratio of clients chosen randomly by the server to participate in the synchronization round t) and T the total number of communication rounds. The authors also propose an algorithm called communication rounds discounting (CRD) that allows the server to adjust, during training, the total number of communication rounds T to an optimal value that leads to achieve a better convergence

performance. Although the authors get important results in terms of controlling the privacy leakage through composition and enhancing the convergence performance, they did not evaluate their contribution against model inversion attacks.

Yu *et al.* [111] use DP to protect the privacy of Internet-of-Things (IoT) devices while collaboratively training an FL model for content popularity prediction. The proposed approach is called FL-based Cooperative Hierarchical Caching (FLCH), it keeps data locally and trains the model using Fog Access Points (F-APs) with their connected IoT devices. The F-AP is responsible for constructing the global model using the weighted averaging method [110] on the gradient received from IoT devices. These IoT devices add a Gaussian noise for protecting their privacy before sending their gradients. However, applying DP is not sufficient to protect the privacy of IoT devices, as an attacker could recover the original data from the noisy gradients sent to the F-AP [9]. Finally, Farokhi *et al.* [131] proved an important result about the relationship between the privacy budget, the size of the dataset, and the loss function. The authors study the cost of privacy for training asynchronously differentially private models with asynchronous communication with different clients. The cost is defined as the mean of a loss function $f(\theta)$ that captures the distance between the output of the ML model $M(x;\theta)$ and the true output y. The authors find that the cost is inversely proportional to the combined size of training datasets squared and the privacy budgets squared.

B. PRIVACY-PRESERVING STATISTICS COLLECTION

LDP consists of applying DP at user devices. The user protects the privacy of his/her data before sending it to the data curator. LDP can be used for collecting data from end-users when users do not trust the data curator. LDP can be implemented either by adding a noise drawn from a probability distribution that satisfies $\epsilon-$DP (such as Laplace, Gaussian, etc.) as stated before in section II, or by implementing the randomized response (RR) [29], [132], [133] technique.

The RR technique consists of flipping the true answer of the user by a certain probability before sending it to the data curator. For example, a social scientist wants to collect statistics from users about drug addiction while maintaining privacy; Before responding to the question, the user toss a coin: 1) if the coin comes up heads then he/she respond truthfully, otherwise 2) the user tosses another coin and respond truthfully if the coin comes up heads, otherwise, the flips his/her response. Specifically, the RR technique is defined as follows:

$$RR(x) = \text{toss a coin b} = \begin{cases} 0, & w.p.\frac{\epsilon}{2}, \\ 1, & \text{otherwise.} \end{cases} \tag{25}$$

- if $b = 0$, then answer truthfully,
- otherwise, if $b = 1$, then toss another coin $\hat{b}$.
  - if $\hat{b} = 0$, then respond truthfully,
  - otherwise, flip the correct response.

RR is proved to satisfy ϵ-DP [132] and it is recently used in many applications [33], [35], [36]. In the rest of this subsection, we explain three practical implementations of ϵ-LDP along with RR by the major technology organizations namely: Apple [36], Microsoft [33], and Google [35].

1) APPLE

Starting from the theory "understanding how people use their devices often helps in improving the user experience", Apple is interested in implementing LDP for their users to understand how they use their devices. It started by studying frequencies per element [36], specifically, estimating typed emojis per web domain; when a user types an emoji, this record (emojis) is privatized via one of the three explained algorithms below and stored locally in the user's device in a list. After a time, the user's system randomly selects some records (i.e., emojis) from the already stored list and sends them to the server. The server, before analysis, strips the privatized records of their IP addresses and any additional sensitive information. The three proposed algorithms for ϵ-LDP are detailed below:

- *Private Count Mean Sketch (CMS) [134]*: outputs a histogram of counts over for a dataset of n records over a domain D. This algorithm is divided into two parts:
  - The client sends his response of size m by mapping the client's response d with one of k hash functions (preliminary defined at the clients and the server). Before sending the response, each bit of the response $v^{(i)}$ is flipped with probability $\frac{1}{1+exp(\epsilon)/2}$ to guarantee the ϵ-differential privacy.
  - The server introduces some noise to the i^{th} client's response $\tilde{v}^{(i)}$ before constructing the sketch matrix M where each row j represents the sum of the users' response who selected the hash function indexed by j. Finally, the server estimates the count for the entry $d \in D$ by debiasing the counts and averaging over corresponding entries in M.
- *Hadamard Count Mean Sketch (HCMS)*: is developed to leverage the cost paid by the clients to send the information to the server. HCMS only requires sending a single bit from the client to the server rather than sending a whole vector as developed in CMS. The client algorithm remains the same, except that here in HCMS the client uses the Hadamard transform to calculate the single bit of the transformed vector w resulting of the client's vector response v. The server algorithm is updated for calculating the sketch matrix M using the transpose matrix H_m^T of H_m used by the client. HCMS gives similar results compared to CMS with the advantage that the client's bandwidth/cost does not scale with the length of the data element d.
- *Private Sequence Fragment Puzzle*: is developed for the case where we don't know the domain elements D, such as counting the most frequent words that the user types and that are not part of any Apple-deployed dictionary.

On the client-side, privacy is guaranteed by applying ϵ privacy budget to the hash function of size 256−bit (used to hash the word) in addition to another ϵ' privacy budget to another sub-string (of length 2 constructed from the original string) add to the original output. The server calculates the frequency oracle $\tilde{f}$ for the word and the frequency oracle $\tilde{f}_\ell$ for the substrings, then it calculates the heavy hitters to get the hash function h and creates the Cartesian product across the hash function and the substrings to create the dictionary.

Although the three proposed algorithms could guarantee strong privacy, they are greedy in terms of resources (i.e., computation and bandwidth). In addition, data elements (i.e., responses) with low frequency may not appear in the statistics at the server.

2) MICROSOFT

LDP does not guarantee strong privacy when we are collecting data repeatedly from the same individuals, such as studying an application usage behavior for several days to improve the user experience. As long as we collect the same statistic, as long as an attacker could learn more information about the real values. For example, if we collect the same statistic T time stamps, the privacy leakage will increase from ϵ to $T \times \epsilon$ (see Theorem 1). Microsoft Research team, Bolin *et al.* [33], handles this problem and proposes new LDP mechanisms for mean and histogram estimation. The first method is called 1−bit LDP mechanism for mean estimation. It is inspired from [135]–[138] with an efficient communication enhancement and stronger protection for repeated data collection. The main idea is based on sending only one-bit $b_i(t)$ at time t for each response $x_i(t)$ that may take the values 0 or m, where $b_i(t)$ is independently drawn from the distribution:

$$b_i(t) = \begin{cases} 1, & \text{with probability } \frac{1}{e^\epsilon+1} + \frac{x_i(t)}{m} \times \frac{e^\epsilon-1}{e^\epsilon+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

Therefore, the data collector calculates an estimate of the mean $\sigma(t)$, for n individuals, as

$$\tilde{\sigma}(t) = \frac{m}{n} \sum_{i=1}^{n} \frac{b_i(t)(e^\epsilon + 1) - 1}{e^\epsilon - 1}. \tag{27}$$

The above mechanism is proved to satisfy ϵ−LDP [33]. Based on the same principle, the authors propose another method called d−Bit mechanism for histogram estimation. In addition, the authors introduce the memoization[7] technique to mitigate privacy leakage for continuously collected statistics. Memoization consists of memorizing the calculated 1−bit response for each specific counter value. At data collection, the client sends the memorized responses without re-calculating the 1−bit responses for already encountered counter values. Hence, an attacker/spy will not learn much information even if he/she collects the client's responses for a very long time. This mechanism has been first implemented in Windows 10 Fall Creators Update to collect the number of seconds that a user has spent using a particular application.

[7]Memoization is an optimization technique used to accelerate calculation by storing the results of expensive functions and returning the cached result for the same inputs.

3) GOOGLE

Earlier, Google [35] has used ϵ-LDP in its proposed algorithm called Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR). RAPPOR is used for privately collecting all types of statistics on clients such as frequencies, histograms, etc. However, the Microsoft approach [33] is less expensive in terms of computation and communications overhead.

In RAPPOR, first, the client's response v is hashed onto a Bloom filter B [139], [140] of size k using a hash function h. Second, each bit i in the Bloom filter B is flipped with a certain probability to get a noisy response vector B':

$$B'_i = \begin{cases} 1, & \text{with probability } \frac{1}{2}p, \\ 0, & \text{with probability } \frac{1}{2}p, \\ B_i, & \text{with probability } 1-p, \end{cases} \tag{28}$$

where p is a user-tunable parameter that controls the level of privacy protection. The resulting new vector B' is memoized and reused for all future response values equal to v. This memoization step is very important to protect the user's privacy when we are collecting the data repeatedly.

Third, the client initializes a new bit array S to 0, and modifies each bit in S with probability:

$$P(S_i = 1) = \begin{cases} q, & \text{if } B'_i = 1, \\ p, & \text{if } B'_i = 0. \end{cases} \tag{29}$$

Fourth, the client sends the new response S to the data curator. It is worth mentioning that steps 3 and 4 allow RAPPOR to guarantee strong privacy protection even for the case where the data is collected repeatedly for a long time. If an attacker gets access to all the individual responses, he/she will be able to only learn the randomized response B' without getting any information about the true response B. Although RAPPOR provides a strong privacy guarantee it is costly in computation and communication overhead. In addition, RAPPOR is not able to detect responses with low frequencies. When the number of different responses increases, their frequencies proportionally decrease and they become hard to detect at low frequencies.

V. OPEN ISSUES AND FUTURE DIRECTIONS

A. COMPOSITION

One of the major shortcomings of differential privacy is that the privacy decreases under composition, we can distinguish two scenarios:

1) **Sequential querying.** The privacy of a fixed pair of dataset neighbors D, D' decreases under the composition of interactive queries; An attacker could learn with some certainty if an individual belongs to a dataset or

not by launching several queries. The composition of k queries each of which is $(\epsilon, \delta)-$differentially private is at least $(k\epsilon, k\delta)-$differentially private [61], [62], [141], [142]. Thus, sequential querying degrades privacy. This issue has been handled by Kairouz *et al.* (Theorem 9) [62] by answering the question: how much privacy is guaranteed after k-fold composition experiment (i.e., after k times databases access). Given k, the authors in [62] define a sequence of privatization mechanisms to guarantee an upper bound on the overall privacy level after the k queries. However, the remaining open questions are about scalability and consistency:

- Does the privacy control leakage, proposed by [62], guarantees strong privacy protection especially when k takes a larger value?
- How does this approach impacts the accuracy?

2) **Stochastic gradient descent.** The privacy protection of a learning model degrades with each stochastic gradient descent iteration. Similar to sequential queries; a composition of k SGD iteration each of which is $(\epsilon, \delta)-$differentially private is at least $(k\epsilon, k\delta)-$differentially private [61], [62]. The released (i.e., trained) learning model becomes crisp against model inversion attacks [81] when the amount of privacy loss (i.e., $k\epsilon$) is large. One of the earliest works that handle this problem is [64], where Abadi *et al.* propose a method called moments accountant (MA) as a tool for tracking the privacy loss across multiple iterations. The MA approach uses Rényi differential privacy [52] in which composition has a simple linear form. In this approach, the privacy budget of the SGD iteration is determined using RDP, afterward, it is mapped back to the standard $(\epsilon, \delta)-$DP by determining ϵ and δ via the relationship between DP and RDP (Theorem 2, [64]). However, this solution is loose, i.e., it does not define an upper bound on the privacy budget. Asoodeh *et al.* [143] derive an approximate of the optimal DP parameters that should guarantee a given level of privacy for about 100 SGD iterations. Although, this approach cannot provide strong privacy beyond 100 iterations, where the greatest need for a solution guaranteeing strong privacy while maintaining a good accuracy regardless of the number of the SGD iterations.

B. EVALUATING DIFFERENTIAL PRIVACY RESISTANCE

Recently, Ren *et al.* [9] propose to use Generative Regression Neural Network (GRNN) for attacking the privacy (i.e., recovering the original dataset) in FL by only using gradients' of clients shared with the server. They found that DP is the most strong approach for protecting privacy. The proposed approach fails to recover the original image when a high level of noise is added to the gradient, however, it succeeds to recover the original image when a small noise (the scale of noise is 0.01) is added to the gradient. Nevertheless, adding a high level of noise leads to poor accuracy. Therefore, the most important question that needs to be answered is: What is the privacy budget ϵ that gives good accuracy while guaranteeing strong privacy protection?

VI. CONCLUSION

In this paper, we provided a detailed survey on differential privacy and its applications. Differential privacy and local differential privacy guarantee strong privacy protection of users' privacy in deep learning, federated learning, and data collection. However, differential privacy still suffers some drawbacks of sequential composition. The privacy degrades as long as the number of composition times increases, which procreate some new variants of $\epsilon-$differential privacy and open new future research directions for tracking the privacy leakage while ensuring a high level of accuracy.

REFERENCES

[1] M. M. Islam, F. Karray, R. Alhajj, and J. Zeng, "A review on deep learning techniques for the diagnosis of novel coronavirus (COVID-19)," *IEEE Access*, vol. 9, pp. 30551–30572, 2021.

[2] F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, "Deep learning in microscopy image analysis: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4550–4568, Oct. 2018.

[3] A. Yahyaoui, A. Jamil, J. Rasheed, and M. Yesiltepe, "A decision support system for diabetes prediction using machine learning and deep learning techniques," in *Proc. 1st Int. Informat. Softw. Eng. Conf. (UBMYK)*, Nov. 2019, pp. 1–4.

[4] S. Blaziunas and A. Raudys, "Comparative study of neural networks and decision trees for application in trading financial futures," in *Proc. Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. (Deep-ML)*, Aug. 2019, pp. 33–38.

[5] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020.

[6] K.-C. Chen, H.-W. Yeh, J.-Y. Hang, S.-H. Jhang, W.-Z. Zheng, and Y.-H. Lai, "A joint-feature learning-based voice conversion system for dysarthric user based on deep learning technology," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 1838–1841.

[7] X. Fei, F. Long, F. Li, and Q. Ling, "Multi-component fusion temporal networks to predict vehicle exhaust based on remote monitoring data," *IEEE Access*, vol. 9, pp. 42358–42369, 2021.

[8] J. H. Abawajy, M. I. H. Ninggal, and T. Herawan, "Privacy preserving social Network data publication," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1974–1997, 3rd Quart., 2016.

[9] H. Ren, J. Deng, and X. Xie, "GRNN: Generative regression neural network—A data leakage attack for federated learning," CoRR, New York, NY, USA, Tech. Rep. 2105.00529, 2021.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[11] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48901–48911, 2019.

[12] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," Tech. Rep., 1998.

[13] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002, doi: 10.1142/S0218488502001648.

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

[15] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 70–78, doi: 10.1145/1401890.1401904.

[16] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, 2006, p. 24.
[17] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115.
[18] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000, doi: 10.1023/A:1026543900054.
[19] R. Canetti, "Security and composition of multiparty cryptographic protocols," *J. Cryptol.*, vol. 13, no. 1, pp. 143–202, Jan. 2000, doi: 10.1007/s001459910006.
[20] R. Canetti, U. Friege, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," Assoc. Comput. Mach., New York, NY, USA, Tech. Rep. 0897917855, 1996.
[21] S. Sayyad, "Privacy preserving deep learning using secure multiparty computation," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 139–142.
[22] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1310–1321, doi: 10.1145/2810103.2813687.
[23] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191, doi: 10.1145/3133956.3133982.
[24] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Symp. Theory Comput. (STOC)*, 2009, pp. 169–178, doi: 10.1145/1536414.1536440.
[25] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generat. Comput. Syst.* vol. 74, pp. 76–85, Sep. 2017.
[26] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.
[27] Q. Zhu and X. Lv, "2P-DNN: Privacy-preserving deep neural networks based on homomorphic cryptosystem," 2018, *arXiv:1807.08459*.
[28] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1351–1362, May 2016.
[29] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Conf. Theory Cryptogr.*, Berlin, Germany: Springer, 2006, pp. 265–284.
[30] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Germany: Springer, 2008, pp. 1–19.
[31] E. Cyffers and A. Bellet, "Privacy amplification by decentralization," CoRR, New York, NY, USA, Tech. Rep. 2012.05326, 2021.
[32] M. Kim, O. Gunlu, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2650–2654.
[33] X. Ding, C. Wang, K.-K. Raymond Choo, and H. Jin, "A novel privacy preserving framework for large scale graph data publishing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 331–343, Feb. 2021.
[34] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, and Y. Yang, "Anonymous and privacy-preserving federated learning with industrial big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6314–6323, Sep. 2021.
[35] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1054–1067, doi: 10.1145/2660267.2660348.
[36] A. Differential Privacy Team. (Dec. 2017). *Learning With Privacy at Scale*. [Online]. Available: https://machinelearning.apple.com/research/learning-with-privacy-at-scale
[37] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" in *Proc. 49th Annu. IEEE Symp. Found. Comput. Sci.*, Oct. 2008, pp. 793–826.
[38] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proc. 22nd ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst. (PODS)*, 2003, pp. 211–222, doi: 10.1145/773153.773174.
[39] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta, "Practical locally private heavy hitters," CoRR, New York, NY, USA, Tech. Rep. 1707.04982, 2017.
[40] F. Mireshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, "Privacy in deep learning: A survey," CoRR, New York, NY, USA, Tech. Rep. 2004.12254, 2020.
[41] T. Ha, T. K. Dang, T. T. Dang, T. A. Truong, and M. T. Nguyen, "Differential privacy in deep learning: An overview," in *Proc. Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov. 2019, pp. 97–102.
[42] A. Boulemtafes, A. Derhab, and Y. Challal, "A review of privacy-preserving techniques for deep learning," *Neurocomputing*, vol. 384, pp. 21–45, Apr. 2020.
[43] H. Dong, C. Wu, Z. Wei, and Y. Guo, "Dropping activation outputs with localized first-layer deep network for enhancing user privacy and data security," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 662–670, Mar. 2018.
[44] N. Papernot, M. Abadi, E. rlingsson. Ú., I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," arXiv, New York, NY, USA, Tech. Rep. 1610.05755, 2017.
[45] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Applications and Techniques in Information Security*, L. Batten, D. S. Kim, X. Zhang, and G. Li, Eds. Singapore: Springer, 2017, pp. 100–110.
[46] M. Xue, C. Yuan, H. Wu, Y. Zhang, and W. Liu, "Machine learning security: Threats, countermeasures, and evaluations," *IEEE Access*, vol. 8, pp. 74720–74742, 2020.
[47] S. Chang and C. Li, "Privacy in neural network learning: Threats and countermeasures," *IEEE Netw.*, vol. 32, no. 4, pp. 61–67, Aug. 2018.
[48] D. Zhang, X. Chen, D. Wang, and J. Shi, "A survey on collaborative deep learning and privacy-preserving," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2018, pp. 652–658.
[49] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, Aug. 2014, doi: 10.1561/0400000042.
[50] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. 24th Annu. Int. Conf. The Theory Appl. Cryptograph. Techn.*, Berlin, Germany: Springer, 2006, pp. 486–503.
[51] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Germany: Springer, 2006, pp. 265–284.
[52] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 263–275.
[53] R. Wang, B. C. M. Fung, and Y. Zhu, "Heterogeneous data release for cluster analysis with differential privacy," *Knowl.-Based Syst.*, vols. 201–202, Aug. 2020, Art. no. 106047.
[54] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," CoRR, New York, NY, USA, Tech. Rep. 1905.02383, 2019.
[55] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," *SIAM J. Comput.*, vol. 41, no. 6, pp. 1673–1693, 2012, doi: 10.1137/09076828X.
[56] F. Fioretto, P. Van Hentenryck, and K. Zhu, "Differential privacy of hierarchical census data: An optimization approach," *Artif. Intell.*, vol. 296, Jul. 2021, Art. no. 103475.
[57] V. Balcer and S. Vadhan, "Differential privacy on finite computers," CoRR, New York, NY, USA, Tech. Rep. 1709.05396, 2019.
[58] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2007, pp. 94–103.
[59] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. B. McMahan, "CPSGD: Communication-efficient and differentially-private distributed SGD," Curran Associates, Red Hook, NY, USA, Tech. Rep. NIPS'18, 2018.
[60] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[61] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Oct. 2010, pp. 51–60.
[62] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, Jun. 2017.
[63] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berkeley Symp. Math. Statist. Probab.*, 1961, pp. 547–561.
[64] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318, doi: 10.1145/2976749.2978318.

[65] S. Asoodeh, J. Liao, F. P. Calmon, O. Kosut, and L. Sankar, "Three variants of differential privacy: Lossless conversion and applications," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 208–222, Mar. 2021.

[66] K. Pan, M. Gong, K. Feng, and K. Wang, "Differentially private regression analysis with dynamic privacy allocation," *Knowl.-Based Syst.*, vol. 217, Apr. 2021, Art. no. 106795.

[67] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012, doi: 10.14778/2350229.2350253.

[68] M. Gong, K. Pan, and Y. Xie, "Differential privacy preservation in regression analysis based on relevance," *Knowl.-Based Syst.*, vol. 173, pp. 140–149, Jun. 2019.

[69] X. Fang, F. Yu, G. Yang, and Y. Qu, "Regression analysis with differential privacy preserving," *IEEE Access*, vol. 7, pp. 129353–129361, 2019.

[70] K. Ligett, S. Neel, A. Roth, B. Waggoner, and Z. S. Wu, "Accuracy first: Selecting a differential privacy level for accuracy-constrained ERM," CoRR, New York, NY, USA, Tech. Rep. 1705.10829, 2017.

[71] J. Lee and C. W. Clifton, "Top-k frequent itemsets via differentially private FP-trees," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 931–940, doi: 10.1145/2623330.2623723.

[72] Z. Sun, Y. Wang, M. Shu, R. Liu, and H. Zhao, "Differential privacy for data and model publishing of medical data," *IEEE Access*, vol. 7, pp. 152103–152114, 2019.

[73] P. Tang, X. Cheng, S. Su, R. Chen, and H. Shao, "Differentially private publication of vertically partitioned data," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 2, pp. 780–795, Mar. 2021.

[74] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, 2012, pp. 638–649, doi: 10.1145/2382196.2382263.

[75] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 1021–1032, Sep. 2010, doi: 10.14778/1920841.1920970.

[76] Y. Xiao, L. Xiong, and C. Yuan, "Differentially private data release through multidimensional partitioning," in *Secure Data Management*, W. Jonker and M. Petković, Eds. Berlin, Germany: Springer, 2010, pp. 150–168.

[77] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor, "Optimizing linear counting queries under differential privacy," in *Proc. 29th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst. Data (PODS)*, 2010, pp. 123–134, doi: 10.1145/1807085.1807104.

[78] D. Huang, S. Han, X. Li, and P. S. Yu, "Orthogonal mechanism for answering batch queries with differential privacy," in *Proc. 27th Int. Conf. Sci. Stat. Database Manage.*, Jun. 2015, pp. 1–10, doi: 10.1145/2791347.2791378.

[79] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2094–2106, Sep. 2014.

[80] H. Huang, D. Zhang, F. Xiao, K. Wang, J. Gu, and R. Wang, "Privacy-preserving approach PBCN in social network with differential privacy," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 931–945, Jun. 2020.

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

[82] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. 23rd USENIX Conf. Secur. Symp. (SEC)*, Berkeley, CA, USA: USENIX Association, 2014, pp. 17–32.

[83] 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. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 2563–2573.

[84] R. Hu, Y. Guo, E. P. Ratazzi, and Y. Gong, "Differentially private federated learning for resource-constrained Internet of Things," CoRR, New York, NY, USA, Tech. Rep. 2003.12705, 2020.

[85] M. Xu and X. Li, "Subject property inference attack in collaborative learning," in *Proc. 12th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Aug. 2020, pp. 227–231.

[86] Y. Qiao, Z. Liu, H. Lv, M. Li, Z. Huang, Z. Li, and W. Liu, "An effective data privacy protection algorithm based on differential privacy in edge computing," *IEEE Access*, vol. 7, pp. 136203–136213, 2019.

[87] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 20–31.

[88] C. Li, M. Hay, G. Miklau, and Y. Wang, "A data- and workload-aware algorithm for range queries under differential privacy," *Proc. VLDB Endowment*, 2014.

[89] W. Qardaji, W. Yang, and N. Li, "Understanding hierarchical methods for differentially private histograms," *Proc. VLDB Endowment*, vol. 6, no. 14, pp. 1954–1965, Sep. 2013.

[90] R. Bassily, "Linear queries estimation with local differential privacy," CoRR, New York, NY, USA, Tech. Rep. 1810.02810, 2018.

[91] Y. Abakarim, M. Lahby, and A. Attioui, "Towards an efficient real-time approach to loan credit approval using deep learning," in *Proc. 9th Int. Symp. Signal, Image, Video Commun. (ISIVC)*, Nov. 2018, pp. 306–313.

[92] W. Wang, J. Lee, F. Harrou, and Y. Sun, "Early detection of Parkinson's disease using deep learning and machine learning," *IEEE Access*, vol. 8, pp. 147635–147646, 2020.

[93] W. K. Al-Jibory and A. El-Zaart, "Edge detection for diagnosis early Alzheimer's disease by using Weibull distribution," in *Proc. 25th Int. Conf. Microelectron. (ICM)*, Dec. 2013, pp. 1–5.

[94] *The Hipaa Privacy Rule*. Accessed: Mar. 25, 2021. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/index.html

[95] *The Pipeda Privacy Law*. Accessed: Mar. 25, 2021. [Online]. Available: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/

[96] T. Wang, N. Li, and S. Jha, "Locally differentially private frequent itemset mining," in *Proc. IEEE Symp. Secur. Privacy, (SP)*, San Francisco, CA, USA, May 2018, pp. 127–143, doi: 10.1109/SP.2018.00035.

[97] M. Maruseac and G. Ghinita, "Precision-enhanced differentially-private mining of high-confidence association rules," *IEEE Trans. Depend. Secure Comput.*, vol. 17, no. 6, pp. 1297–1309, Nov. 2020.

[98] D. Su, J. Cao, N. Li, and M. Lyu, "PrivPfC: Differentially private data publication for classification," *VLDB J. Int. Very Large Data Bases*, vol. 27, no. 2, pp. 201–223, Apr. 2018, doi: 10.1007/s00778-017-0492-3.

[99] Y. Zhang, Z. Hao, and S. Wang, "A differential privacy support vector machine classifier based on dual variable perturbation," *IEEE Access*, vol. 7, pp. 98238–98251, 2019.

[100] N. L. Zhang, "Hierarchical latent class models for cluster analysis," *J. Mach. Learn. Res.*, vol. 5, pp. 697–723, Dec. 2004.

[101] N. Mohammed, D. Alhadidi, B. C. M. Fung, and M. Debbabi, "Secure two-party differentially private data release for vertically partitioned data," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 1, pp. 59–71, Jan. 2014.

[102] R. J. Lewis, "An introduction to classification and regression tree (CART) analysis," in *Proc. Annu. Meeting Soc. Academic Emergency Med.*, 2000, pp. 1–14.

[103] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1994, pp. 487–499.

[104] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, 2000, doi: 10.1145/335191.335372.

[105] R. Chen, B. C. M. Fung, and B. C. Desai, "Differentially private trajectory data publication," CoRR, New York, NY, USA, Tech. Rep. 1112.2020, 2011.

[106] F. McSherry and R. Mahajan, "Differentially-private network trace analysis," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 123–134, Aug. 2010, doi: 10.1145/1851275.1851199.

[107] R. Chen, B. C. M. Fung, P. S. Yu, and B. C. Desai, "Correlated network data publication via differential privacy," *VLDB J.*, vol. 23, no. 4, pp. 653–676, Aug. 2014, doi: 10.1007/s00778-013-0344-8.

[108] X. Ying and X. Wu, "Randomizing social networks: A spectrum preserving approach," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 739–750, doi: 10.1137/1.9781611972788.67.

[109] B. Bebensee, "Local differential privacy: A tutorial," CoRR, New York, NY, USA, Tech. Rep. 1907.11908, 2019.

[110] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, vol. 54. Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[111] Z. Yu, J. Hu, G. Min, Z. Wang, W. Miao, and S. Li, "Privacy-preserving federated deep learning for cooperative hierarchical caching in fog computing," *IEEE Internet Things J.*, early access, May 18, 2021, doi: 10.1109/JIOT.2021.3081480.
[112] L. Yin, J. Feng, H. Xun, Z. Sun, and X. Cheng, "A privacy-preserving federated learning for multiparty data sharing in social IoTs," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2706–2718, Jul. 2021.
[113] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, and H. V. Poor, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Trans. Mobile Comput.*, early access, Feb. 4, 2021, doi: 10.1109/TMC.2021.3056991.
[114] Y. Liu, R. Zhao, J. Kang, A. Yassine, D. Niyato, and J. Peng, "Towards communication-efficient and attack-resistant federated edge learning for industrial Internet of Things," ACM Trans. Internet Technol., New York, NY, USA, Tech. Rep. 1533-5399, 2020.
[115] A. Sonee and S. Rini, "Efficient federated learning over multiple access channel with differential privacy constraints," CoRR, New York, NY, USA, Tech. Rep. 2005.07776, 2020.
[116] H. Wu, C. Chen, and L. Wang, "A theoretical perspective on differentially private federated multi-task learning," 2020, *arXiv:2011.07179*.
[117] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5827–5842, Jul. 2020.
[118] M. Gong, J. Feng, and Y. Xie, "Privacy-enhanced multi-party deep learning," *Neural Netw.*, vol. 121, pp. 484–496, Jan. 2020.
[119] A.-T. Tran, T.-D. Luong, J. Karnjana, and V.-N. Huynh, "An efficient approach for privacy preserving decentralized deep learning models based on secure multi-party computation," *Neurocomputing*, vol. 422, pp. 245–262, Jan. 2021.
[120] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multi-party computing," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6178–6186, Apr. 2021.
[121] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2021, *arXiv:1912.04977*.
[122] X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton, "A methodology for formalizing model-inversion attacks," in *Proc. IEEE 29th Comput. Secur. Found. Symp. (CSF)*, Jun. 2016, pp. 355–370.
[123] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 225–240, doi: 10.1145/3319535.3354261.
[124] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The limitations of federated learning in Sybil settings," in *Proc. 23rd Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*. San Sebastian, Spain: USENIX Association, Oct. 2020, pp. 301–316. [Online]. Available: https://www.usenix.org/conference/raid2020/presentation/fung
[125] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. USENIX Secur.*, Aug. 2020, pp. 1605–1622.
[126] Y. Zhang and Q. Yang, "A survey on multi-task learning," CoRR, New York, NY, USA, Tech. Rep. 1707.08114, 2018.
[127] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," CoRR, New York, NY, USA, Tech. Rep. 2009.09796, 2020.
[128] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "HybridAlpha: An efficient approach for privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur. (AISec)*, 2019, pp. 13–23, doi: 10.1145/3338501.3357371.
[129] A. Triastcyn and B. Faltings, "Federated learning with Bayesian differential privacy," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2587–2596.
[130] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. 26th USENIX Conf. Secur. Symp.*, Berkeley, CA, USA: USENIX Association, 2017, pp. 729–745.
[131] F. Farokhi, N. Wu, D. Smith, and M. A. Kaafar, "The cost of privacy in asynchronous differentially-private machine learning," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2118–2129, 2021.
[132] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *J. Mach. Learn. Res.*, vol. 17, pp. 492–542, Jan. 2016.
[133] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *J. Amer. Statist. Assoc.*, vol. 60, no. 309, pp. 63–69, 1965.
[134] G. Cormode, *Count-Min Sketch*. Boston, MA, USA: Springer, 2009, pp. 511–516.
[135] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proc. 47th Annu. ACM Symp. Theory Comput.*, Jun. 2015, pp. 127–135, doi: 10.1145/2746539.2746632.
[136] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, Oct. 2013, pp. 429–438.
[137] J. C. Duchi, M. Jordan, and M. J. Wainwright, "Local privacy and minimax bounds: Sharp rates for probability estimation," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Red Hook, NY, USA: Curran Associates Inc., 2013, pp. 1529–1537.
[138] J. Duchi, M. Wainwright, and M. Jordan, "Minimax optimal procedures for locally private estimation," arXiv, New York, NY, USA, Tech. Rep. 1604.02390, 2017.
[139] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970, doi: 10.1145/362686.362692.
[140] G. Bianchi, L. Bracciale, and P. Loreti, "'Better than nothing' privacy with Bloom filters: To what extent?" in *Privacy in Statistical Databases*, J. Domingo-Ferrer and I. Tinnirello, Eds. Berlin, Germany: Springer, 2012, pp. 348–363.
[141] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Germany: Springer, 2006, pp. 486–503.
[142] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. 41st Annu. ACM Symp. Symp. theory Comput. (STOC)*, 2009, pp. 371–380, doi: 10.1145/1536414.1536466.
[143] S. Asoodeh, J. Liao, F. P. Calmon, O. Kosut, and L. Sankar, "Three variants of differential privacy: Lossless conversion and applications," *IEEE J. Sel. Areas Inf. Theory*, 2021.

AHMED EL OUADRHIRI received the B.S., M.S., and Ph.D. degrees in computer science from Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2010, 2012, and 2017, respectively. From 2017 to 2018, he was a Research Assistant with Alfaisal University, Riyad, Saudi Arabia. From 2019 to 2020, he was a Technical Project Manager at Carta Worldwide, Casablanca, Morocco. He is currently a Postdoctoral Fellow with the Department of Engineering Technology, University of Houston, Houston, TX, USA. His research interests include differential privacy, deep learning, federated learning, and networks and systems.

AHMED ABDELHADI (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, in 2011. He is currently an Assistant Professor with the University of Houston. Before joining UH, he was a Research Assistant Professor at Virginia Tech. He was a member of the Wireless Networking and Communications Group (WNCG) and the Laboratory of Informatics, Networks and Communications (LINC) Group during his Ph.D. In 2012, he joined the Bradley Department of Electrical and Computer Engineering and the Hume Center for National Security and Technology, Virginia Tech. He was a Faculty Member of Wireless @ Virginia Tech. He has coauthored more than 90 journals and conference papers and seven books in these research topics. His book *Cellular Communications Systems in Congested Environments* is bookplated as the Virginia Tech Provost's Honor Book and his book *Resource Allocation with Carrier Aggregation in Cellular Networks* is featured in the 13th Annual Virginia Tech Authors Recognition Event. His research interests include the areas of wireless communications and networks, artificial intelligence, cyber-physical systems, and security. He received the Silver Contribution Award from IEEE International Conference on Computing, Networking and Communications (ICNC), the Best Paper Award from IEEE International Symposium on Systems Engineering (ISSE), and the Outstanding Paper Award from IEEE International Conference of Advanced Communications Technology (ICACT).