Differentially Private Federated Learning: Privacy and Utility Analysis of Output Perturbation and DP-SGD.

Anastasia Pustozerova
SBA Research, Vienna, Austria
apustozerova@sba-research.org

Rudolf Mayer
SBA Research, Vienna, Austria
rmayer@sba-research.org,

***Abstract*—Federated learning is a technique that enables multiple parties to train a machine learning model collaboratively from data already residing in different locations, e.g. data silos. Instead of aggregating the private data from the silos to a central place, federated learning requires only exchanging and aggregating the machine learning models. These models are locally trained by the parties on their private data, which thus never leaves the silo. However, the models may still leak sensitive information about the training data in the form of e.g. membership disclosure. To mitigate these residual privacy risks in federated learning, one has to use additional defence techniques such as Differential Privacy (DP), which introduces noise into the training data or the model. Differential Privacy provides a mathematical definition of privacy and can be applied in machine learning via different perturbation mechanisms. This work focuses on the analysis of Differential Privacy in federated learning through (i) output perturbation of the trained machine learning models and (ii) a differentially-private form of stochastic gradient descent (DP-SGD). We consider these two approaches in various settings and analyse their performance in terms of model utility and achieved privacy. To evaluate a model's privacy risk, we empirically measure the success rate of a membership inference attack. We observe that DP-SGD allows for a better trade-off between privacy and utility in most of the considered settings. In some settings, however, output perturbation is able to provide a better or similar privacy-utility trade-off and at the same time better communication and computational efficiency.**

***Index Terms*—Federated Learning, Differential Privacy, Output Perturbation, DP-SGD**

I. Introduction

Data used to train machine learning (ML) models is often distributed among different entities, e.g. at various healthcare providers, mobile phones or IoT devices and has to be collected at a centralised place for processing. In some scenarios, aggregating data in one place may not be possible due to regulatory or technical constraints, or the desire of the data owners to preserve the privacy of their data. Federated learning (FL) enables the training of machine learning models on distributed data without the need for sharing sensitive training data with other parties. In FL, the participants (also called clients or nodes) train models locally on their data and share only the models' updates with an aggregator. The aggregator (or federated server) receives all locally trained machine learning models and averages them into a global model, using e.g. the *federated averaging* algorithm [1].

While enhancing clients' data privacy is one of the main motivations behind federated learning, there are still risks that might threaten the privacy of this data. The models, which are trained on private data and shared in FL, represent an abstraction of the training data and, as was shown in several works, are prone to data leakages [2]. Malicious actors with access to models trained in FL can perform attacks to infer sensitive information about the training data [3]. These attacks include e.g. membership inference [4], or data reconstruction attacks like model inversion [5], or a similar attack through the gradient leakage [6]. Considering these residual privacy risks in federated learning and to guarantee stronger privacy, one should employ additional mitigation mechanisms. Popular privacy-enhancing techniques in federated learning include cryptographic approaches like Homomorphic Encryption (HE) or Secure Multi-Party Computation (SMPC), and Differential Privacy (DP) (see Section II for details). One can use either of these three approaches or a combination of them, to mitigate different privacy risks in federated learning.

In this work, we focus on the analysis of Differential Privacy, as this technique can mitigate privacy risks caused by various types of attackers (see Section II). By introducing noise to the locally trained models, DP allows clients to protect their private data from leaking information through local and global models in FL, therefore mitigating privacy risks coming not only from malicious clients, for which HE and SMPC can be used as a defence too, but also from the malicious users of intermediate and final global models.

One can use DP methods at different stages of the machine learning process, e.g. a differentially-private form of stochastic gradient descent (DP-SGD), output or objective perturbation (see Section III). *DP-SGD* [7] is arguably the most popular approach for applying DP in machine learning, as it can be used on the wide range of machine learning models that are trained with the SGD optimiser, e.g. neural networks.

This work received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078 (FeatureCloud). This publication reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains. SBA Research (SBA-K1) is a COMET Center within the COMET - Competence Centers for Excellent Technologies Programme and funded by BMK, BMAW, and the federal state of Vienna. The COMET Programme is managed by FFG.

Meanwhile, *output perturbation* [8] is less explored, as its application is limited to machine learning models with known sensitivity bound e.g. logistic regression or support vector machine (SVM) (see Section IV). However, output perturbation is more efficient and easier to implement, as it requires adding noise to the trained models, while in DP-SGD the noise is generated and added during training at each iteration.

After analysing existing works and identifying the gaps in the current research of DP in federated learning, we focus on achieving DP through output perturbation in FL, and contrast it to the DP-SGD algorithm applied in FL. We analyse which of the considered approaches results in a better privacy-utility trade-off. We perform a Membership Inference Attack (MIA) to empirically measure the privacy loss of models shared in FL, and analyse how it corresponds to the privacy leakage parameter ϵ used in DP. We present a comprehensive experimental analysis of output DP and DP-SGD in various federated settings and then compare these two approaches in terms of:

- **Effectiveness**: utility of the resulting global model in FL;
- **Communication efficiency**: the number of FL iterations (rounds of models communication) required to train an effective global model;
- **Privacy leakage**: leakage of clients' data through locally trained models in FL, measured by membership inference attack accuracy on the models shared in FL.

Conducting a comprehensive experimental evaluation of output perturbation and DP-SGD in various federated learning settings, we find that:

- DP-SGD exhibits a better trade-off between privacy and utility than output perturbation in most of the considered FL settings. However, for one of the considered datasets, when data is equally distributed among the clients in FL, output perturbation provides a better or similar privacy-utility trade-off as DP-SGD, while requiring fewer FL iterations for the global model to converge. Being more communication and computationally efficient than DP-SGD, output perturbation would be preferable to use in such cases.
- Local models tend to leak more information about training data in the first FL iterations. Therefore, the recommendation is to protect them better by using smaller ϵ (privacy budget parameter) in the first few iterations of federated learning.
- Both output perturbation and DP-SGD have a larger impact on the privacy and utility of the models with the larger number of nodes in the FL settings (more than eight nodes).
- The same ϵ leads to different empirical privacy leakage for output perturbation compared to DP-SGD and also depends on the dataset and FL setting. There is no universally "good" epsilon – the privacy budget parameter has to be tuned for specific machine learning tasks, datasets and a used DP method.

The remainder of this paper is organised as follows. Section II defines the threat model and considers mitigation strategies in federated learning. In Section III, we discuss different techniques for achieving DP in machine learning, and existing works on DP in federated learning. Section IV describes the DP mechanisms analysed in the current work. In Section V, we detail our experimental setup for reproducibility purposes. before we discuss and analyse the main findings from the extensive experimental evaluation of DP in federated learning in Section VI. We provide conclusions and an outlook on future work in Section VII.

II. Threat Model and Mitigation Strategies in Federated Learning

We distinguish three different attackers in federated learning, based on their goals and knowledge:

1) **Malicious server** (or other attackers obtaining that access) having access to the local models. The attack targets are thus the local models, and the goal for an attacker in this scenario is to infer sensitive information about training data from the local models, which are shared in federated learning. Therefore, the privacy of clients' data in federated learning might be at risk.
2) **Malicious client** (or other users) having access to the intermediate global models. The attack target is a global model in its intermediate (after each FL iteration) and final state. The goal of an attacker is to infer sensitive information about the training data of other FL participants from the global model. An attacker can have knowledge about training parameters and have white-box access to the global model and some of the local models.
3) **Malicious users of a final global model**. The attack target is the final (resulting) global model, which is trained with federated learning and can be shared with other parties for usage. In this scenario, the goal of the attack is to infer information from the final global model. An attacker can have white-box or black-box access to the target model.

All considered attackers in federated learning target the same goal – inference of the sensitive information about clients' training data. The attackers however differ in their knowledge and model access. An attacker that has only access to the global model has fewer chances to perform successful MIA, as the global model is an averaged combination of local models. A malicious server, on the other hand, is the most dangerous attacker, as they have access to the local models directly.

To reduce privacy risks in federated learning, one can apply some or a combination of the following, frequently used mitigation strategies:

a) Homomorphic encryption: (HE) [9]. In federated learning, clients can encrypt their models before sending them to the aggregation server. The server then performs computations on this encrypted data (e.g. federated averaging) without being able to decrypt it. The server then sends the output of the computation, i.e. the global model, to the clients – the only parties that can decrypt and use the output. Therefore,

HE can be used to mitigate privacy risks in the case of a **malicious server**. The clients need to agree on key exchange to facilitate encryption of the local models' weights. One of the main problems with applying HE is the computational overhead that is caused by the encryption and decryption processes. This can prohibitively reduce the efficiency of the whole federated learning training, especially with large neural networks [10].

b) Secure multi-party computation: (SMPC) [11] can be used in federated learning to securely compute the average of the models shared during federated training, providing protection against **malicious server**, similar to HE. SMPC is a cryptographic protocol that allows participants to jointly compute a public function (e.g. averaging) over their private data (local models' weights). In SMPC, the model weights are not accessible to any party, besides their owner. The main drawback of SMPC is low efficiency, as it requires a significant amount of additional communication between the clients.

c) Differential Privacy: (DP) provides a mathematical definition of privacy, by introducing a level of uncertainty into the model. In machine learning, Differential Privacy can be applied by adding noise e.g. to the training data, trained model, gradients or objective function (for more explanation about DP see Section IV). DP can be applied before, during, or after the training to ensure the privacy of the resulting output (the model). This property makes DP a versatile solution that can be used to secure from different types of attackers including the **malicious server, client and user of the final global model**. In fact, DP is the only of our discussed mitigation techniques that can be used to defend against inference attacks on the output of the FL training process, i.e. global model. Nevertheless, one of the main drawbacks of DP is its effect on machine learning model effectiveness. The noise added in Differential Privacy inevitably causes a drop in the utility of the model. Therefore, when applying Differential Privacy, one always has to acknowledge the trade-off between the model's privacy and utility.

In this work, we focus on the DP approach, as it allows for mitigating against different types of attackers. Differential Privacy also enables quantifying privacy loss by a **privacy budget parameter -** ϵ. The lower the ϵ, the less the leakage from a differentially private machine learning model. Therefore, the party training and contributing a machine learning model in federated learning has an instrument to calibrate the privacy level of the model that is acceptable to them.

III. Related work

Differential Privacy (DP) provides a formal mathematical definition of privacy and de facto became a standard for analysing privacy leakage [12]. DP was defined by Dwork et al. [13] to secure a database containing sensitive information, while being able to query statistics about the data. They introduced a privacy budget parameter ϵ which one can use to regulate the privacy level. Further, they showed several critical properties of DP, such as sequential composition [14] (see Section IV-C for more details). They also introduced the Laplace and Gaussian mechanisms, enabling the calculation of added noise in DP [14].

One of the first to apply DP mechanisms in machine learning were Chaudhuri et al. [15]. They presented *output perturbation* for training a privacy-preserving regularised Logistic Regression classifier. The approach is based on the sensitivity method from [13]. It allows adding noise to a trained Logistic Regression model and guarantees that it is differentially private. Moreover, they present a new algorithm to train privacy-preserving classifiers - *objective perturbation*. In objective perturbation, the noise is added to the objective function during training. Later, Chaudhuri et al. [8] extend their work and show how output perturbation and objective perturbation can be applied to regularised Empirical Risk Minimisation and Support Vector Machine. For classifiers using stochastic gradient descent (SGD) to optimise a loss function, Song et al. [16] introduced the differentially private stochastic gradient descent (DP-SGD). Later, Abadi et al. [7] extended their approach and suggested a new method of privacy budget accounting, which allowed for reducing the amount of noise added in DP. DP-SGD became a widely used approach to train differentially private machine learning models and was implemented in many privacy libraries [12].

In federated learning, DP can be applied at different stages of the training or communication, depending on the requirements of data models and a threat model:

- **Central Differential Privacy** is applied when users trust the data aggregator, which in turn applies DP only on the global model to protect data privacy when the global model is shared for public usage.
- **Local Differential Privacy** [17] refers to the case when an aggregator cannot be trusted and each party wants to protect their data (local models) before sending it to the aggregator (e.g. performing input perturbation or training of differentially-private local models). As local DP is stricter than central DP, it usually results in a more significant drop in the utility of the model. Truex et al. [18] consider local DP in federated learning with neural networks and suggest a novel approach allowing clients in FL to train complex models. They, however, achieve only Condensed Local Differential Privacy (CLDP), which is a relaxation of ϵ-DP. The approach is based on two steps: perturbation of complex models' parameters and selective sharing of these parameters at different FL iterations. Sun et al. [19] propose a mechanism to achieve local DP when training neural networks in federated learning. The method is based on adapting to the different model weights' ranges and parameter shuffling to make it harder to find out from which client the updates came to the aggregator.
- **Distributed Differential Privacy** aims to achieve the utility of central DP, but without having to trust the central aggregator. Distributed DP can be implemented by using e.g. secure aggregation protocols like SMPC or Homomorphic Encryption to protect the confidentiality of the model parameters from the aggregator. At the

same time, clients can apply local DP in a manner that after aggregation the global model will have the same amount of noise as in central DP. In [20], the authors suggested a differentially private FL system which allows achieving distributed DP, when the sum of the clients' local models is a DP function, and original local models are protected by secure aggregation protocol. Jarin et al. [21] use SMPC to secure local models and add DP noise to the global model to secure it from the inference attacks of malicious clients.

- **Hybrid Differential Privacy** [22] considers scenarios when different clients have different privacy requirements or restrictions: while some of them may desire to have local DP guarantees, for others central DP or no DP at all is a viable option. In this case, the utility of the global model can be significantly improved.

In [23], the authors introduced DP-Federated Averaging and DP-Federated SGD algorithms based on the idea from DP-SGD suggested by [7]. Randomly sampling clients at each FL iteration allows using the moment accountant method from [7] (as randomly sampling instances) to provide a tighter bound on the privacy loss for the whole federated learning computation. The clients in FL locally train the model using DP-SGD and send differentially private gradients to the server for aggregation. In the experimental evaluation, they show that given a sufficiently large number of clients in FL, DP does not result in significant utility loss, but rather comes at the cost of increased computation.

Jarin et al. [24] provide an analysis of Differential Privacy in a centralised setting, considering input perturbation, output perturbation, objective perturbation, gradient perturbation and prediction perturbation approaches. Following their work, we provide a comprehensive analysis of Differential Privacy in a federated learning setting. We consider output and gradient perturbation approaches for achieving local differential privacy in different settings, including non-independent and identically distributed (non-IID) data. Many related works (e.g. [18], [19], [23]) focus on measuring only the utility loss when applying DP in FL and try to optimise the privacy-utility trade-off where privacy is defined by the ϵ parameter. In our work, we also consider the empirical privacy loss measured by the success rate of a membership inference attack. We show that the same values of ϵ can correspond to very different empirical privacy risks, depending on the dataset and model characteristics. By conducting an extensive experimental evaluation, we assess which DP technique provides a better privacy-utility trade-off in federated learning.

IV. Differential Privacy

Consider a function f mapping a database to reals $f : D \rightarrow \mathbb{R}$. In machine learning, that function represents a machine learning algorithm. Dwork et al. [13] proved that the privacy of the database can be preserved by adding noise according to the *sensitivity* of the function f. Essentially, the **sensitivity** of f denotes the maximum possible impact on the output of the function f, caused by removing or adding any single instance to the database.

A. Differential Privacy via Output Perturbation

Output perturbation (short: Output DP) refers to the method of modifying an already trained model's weights (θ). To get a differentially private model, noise is added to this trained model: $\theta_{dp} = \theta + noise$. We use the Gaussian mechanism [14] to add noise sampled from a Gaussian distribution. The Gaussian mechanism guarantees (ϵ, δ)-Differential Privacy, which is a relaxation of ϵ-DP, where δ is a parameter that controls the strength of relaxation. In the Gaussian mechanism, the $noise$ is sampled from a normal distribution $N(0, \sigma^2)$, where $\sigma = S(f;2)\sqrt{2ln(1.25/\delta)}/\epsilon$ [14], $S(f;2)$ denoting the $l2-$sensitivity of the model. Chaudhuri et al. [15] proved that the sensitivity of Logistic Regression with a regularisation parameter λ is at most $\frac{2}{n\lambda}$, where n is the number of samples in the database. This allows the development of a privacy-preserving Logistic Regression algorithm based on the sensitivity approach. Finding the bound to the sensitivity is only possible for simpler models, as more complex models have complex relations between input and output [25].

B. Differentially Private Stochastic Gradient Decent

Differentially Private Stochastic Gradient Decent (DP-SGD) [7] allows training a differentially private machine learning model by injecting noise during the training. DP-SGD adds two additional steps to the original mini-batch SGD algorithm:

1) When computing the gradient for the mini-batch of samples from the original dataset, clip the l_2 norm of each per-example gradient $g(x_i)$, where x_i is an instance from the selected mini-batch and C is a gradient norm bound:
$$g(x_i) \leftarrow g(x_i)/max(1, \frac{||g(x_i)||_2}{C})$$
2) Add noise to the aggregated gradient of the batch:
$$\overline{g} \leftarrow \frac{1}{L}(\sum_i g(x_i) + N(0, \sigma^2 C^2 \mathbf{I})),$$
where L is a mini batch size, $\mathbf{I}$ is an identity matrix and σ is a noise scale. Compute the gradient update based on the noised gradient $\overline{g}$:
$$\theta_{t+1} \leftarrow \theta_t - \alpha\overline{g},$$
where α is the learning rate and t is the iteration number.

DP-SGD is a widely used approach to achieve Differential Privacy for machine learning models, as it, unlike output DP, does not require knowledge of the model sensitivity.

C. Parallel and Sequential compositions

DP-SGD is a composition of t Gaussian mechanisms, which makes it (ϵ, δ)-differentially private. Composition is an important property of Differential Privacy. **Sequential Composition** guarantees that the application of multiple DP mechanisms on the same database is still differentially private [14]. For

the combination of several (ϵ_i, δ_i)-DP mechanisms applied on the same dataset, the privacy loss ϵ is calculated as the sum of privacy losses of each (ϵ_i, δ_i)-DP mechanism: $\epsilon = \sum_i \epsilon_i$ and $\delta = \sum_i \delta_i$. DP-SGD, for example, uses the sequential composition property to guarantee Differential Privacy and compute the privacy loss of SGD after multiple iterations.

Parallel composition allows computing the privacy loss of the DP mechanisms applied on disjoint datasets. The Privacy loss of a combination of several (ϵ_i, δ_i)-DP mechanisms, applied on disjoint datasets, is the maximal privacy loss from all the (ϵ_i, δ_i)-DP mechanisms: $\epsilon = max(\epsilon_i)$ and $\delta = max(\delta_i)$. Parallel composition allows computing the privacy loss in federated learning when performing federated averaging after the first federated learning iteration. The privacy loss for the global model after the first aggregation is equal to the highest privacy loss out of all local models. However, as of the second iteration of federated learning, local models are trained based on the global model, one cannot assume the independence of the local models. Therefore, applying parallel composition is no longer possible.

D. Privacy Budget in Federated Learning

In federated learning, we use the sequential composition theorem to calculate the privacy loss for the local models after several federated learning iterations. In our case, we consider the clients to have the same privacy loss, therefore, the global model is also (ϵ_i, δ_i)-differentially private due to the parallel composition. The clients get the global model and proceed to optimise it on the local data, applying again (ϵ_i, δ_i)-DP mechanisms.

V. Experimental setup

To ensure the reproducibility of our work, we provide a thorough description of the experimental setup, the datasets preprocessing and the source code[1].

A. Datasets

For the experimental evaluation, we use two datasets:

Purchase-100 (Purchase) dataset is frequently used in works carrying out a membership inference attack, as it was introduced by Shokri et al. in the MIA-defining paper [4]. We utilise the preprocessed version of the dataset[2] (for the preprocessing steps and the original data, see [4]). The preprocessed version of the dataset contains almost 200K samples, representing customers, where 600 binary attributes describe whether they were buying a specific product or not. The classification task is to determine the purchase behaviour group for each customer. There are 100 different groups (i.e. 100 classes in a classification task). Similar to the experiment setup by [4], for our empirical evaluation, we use 10K randomly selected samples for training, 2K for validation (hyperparameters tuning), 10K for testing and the rest for building shadow models for MIA. Using Logistic Regression, we achieve an accuracy score of 0.56 with a learning rate of 0.001, l2-regularisation of $1e-4$ and 50 iterations. Shokri et al. [4] achieve an accuracy of 0.67 with neural network and 0.504 using the Amazon ML-as-a-service platform.

LendingClub-Loan (Loan) dataset was used in a recent work evaluating Differential Privacy approaches in machine learning in a centralised setting [24]. We obtain the LendingClub-Loan dataset from Kagle[3] and preprocess the dataset using the Juypter notebook from the [24]. The dataset contains information about borrowers and the loans they want to take. The classification task is to determine one of the six risk groups, based on which the bank defines the interest rate for the client. The full dataset contains 100K samples. To be in line with the Purchase dataset, we also use randomly selected 10K samples for training, 2K for validation (hyperparameters tuning), 10K for testing and the rest for building shadow models for MIA. The baseline accuracy score for the centralised setting is 0.86, with a learning rate of 0.01, l2-regularisation of $1e-6$ and 200 iterations. In [24], the authors do not report accuracy, but only utility loss, however from their code available on GitHub, we find that our model achieves almost 10% higher accuracy than in [24].

B. Differential Privacy

To find optimal hyper-parameters in the setting with DP, we use grid search and find that in order to achieve higher effectiveness of the global model when using DP-SGD, each client in federated learning has to train the model locally with a larger number of epochs, compared to no DP case. To achieve a centralised baseline accuracy score for the Purchase dataset, we need to increase the number of iterations from 50 (without DP) to 200. This happens due to the dependency of the amount of noise on the number of iterations in SGD. To achieve the best performance with DP-SGD we also tuned the mini-batch size and for final evaluation used 20 samples for both Purchase and Loan datasets. After the grid search, the norm bound in DP-SGD was set to 2 for the Purchase dataset and 10 for the Loan dataset. For output Differential Privacy, we tuned the l2-regularisation parameter: for both Purchase and Loan datasets, we use l2-regularisation of $1e-4$ when applying output DP. The recommendation in literature for the parameter δ is to use $\delta << 1/n$, where n is the number of samples [12]. As both of our datasets contain 10K samples, we use $\delta = 1e-5$.

C. Membership Inference Attack

The Membership Inference Attack (MIA) is widely used in privacy-preserving machine learning research to estimate (and compare) privacy leakage. The goal of an attacker performing MIA is to infer whether some particular sample was used for training the *target* machine learning model. Therefore, the *membership of a sample in the training set* is the sensitive information that an attacker is aiming to infer, from having access to the target model.

We use *attack models* based on *shadow models* as described in [4]. To train different attack models, we vary the number of

[1]https://github.com/sbaresearch/Differential_Privacy_in_Federated_Learning

[2]https://www.comp.nus.edu.sg/~reza/files/datasets.html

[3]https://www.kaggle.com/datasets/wordsforthewise/lending-club

shadow models (1,5,10), shadow models training set size (10K, 15K, 20K) and attack models hyper-parameters (learning rate: [0.01, 0.001, 0.0001] and epochs number: [100,200,500]). From all the attack models (trained with different parameter combinations), we select five of the best-performing attack models for each dataset. For the shadow models, we used Logistic Regression with the same hyper-parameters as the target model. For the attack models, we chose a neural network with one hidden layer of 64 neurons and a $ReLU$ activation function similar to [4]. All the target models in federated settings, i.e. the local and global models, were attacked by five attack models. In the empirical evaluation (see Section VI), MI attack accuracy represents the mean attack accuracy from the 5 attack models. The attack model represents a binary classifier which predicts if some particular instance was in the training set of the model, or not. Therefore, the higher the attack accuracy - the higher the privacy leakage of the model. The attack model test set always contains 50% samples that were in the training data of the target model and 50% of samples that were not used for training the target model. Therefore, the baseline for attack accuracy is 0.5, representing the accuracy of random guessing. In the Figures presented in Section VI, we denote this baseline as "no_privacy_leakage"

D. Federated Learning Setup

We consider a federated learning setting with 2, 4, 8, 16 and 32 nodes (or clients). We consider IID and non-IID data distributions. In the first case, the whole training dataset is randomly and evenly distributed between the clients. In the second case, we simulate a quantity skew in the federated setting. For computing the global model, we use federated averaging algorithms, where we average the weights of the local models to get a global model [1]. Each client has the same hyper-parameters for training local models.

E. Evaluation Metrics

The main goal of the current work is to find the DP strategy in federated learning that results in the best privacy-utility trade-off. Therefore, during the evaluation, we especially focus on metrics like the accuracy and utility loss of the global model, and the MIA accuracy on the local models.

The global model accuracy shows if federated learning is a useful solution for the considered classification task. In federated learning, the global model should achieve higher performance in terms of effectiveness than the local models. Ideally, the effectiveness of the global model should be close to a model trained on the centralised data – through a centralised model is often not possible due to e.g. data protection regulations. Still, in the evaluation of the experiment, we compare the accuracy of the global models in federated learning to the centralised baseline (see Section V-A) as an upper bound. We also use utility loss to evaluate the effect of DP on models' effectiveness. Utility loss is defined by the difference between the highest reached global model accuracy in the FL setting with the corresponding number of nodes without DP and global model accuracy with output perturbation or DP-SGD approach and different epsilon values.

The second metric helps us to analyse empirical privacy risks in federated learning. The privacy budget parameter ϵ is used in DP to regulate privacy leakage. We evaluate how ϵ corresponds to the empirical privacy leakage measured in the accuracy of membership inference attacks. We especially focus on attacks on the local models in federated learning, as we consider a threat model with a malicious aggregator (see Section II) and these local models tend to leak more information about training data than global models.

VI. Results and Discussion

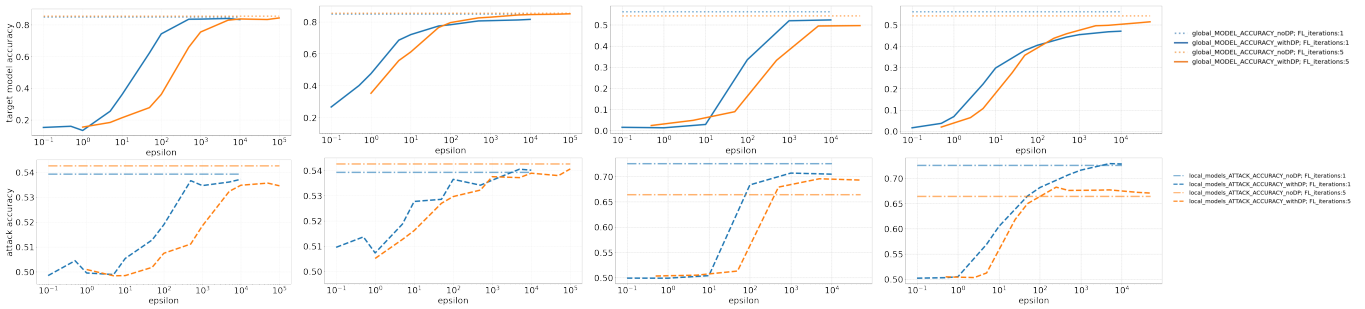
In the following section, we will provide results and discuss several aspects of our experimental evaluation. We start with evaluating the impact of DP on the efficiency of an FL process by analysing how many FL iterations is needed for a global model to converge with different privacy budget. We then discuss the effect of having different numbers of nodes in the federation, before we specifically contrast the DP-SGD and Output DP approaches to show which of them provides a better privacy-utility trade-off. Finally, we also investigate settings where the amount of training data in each node is different (non-IID) and how that affects the privacy and utility trade-off when using output perturbation and DP-SGD.

A. Federated Learning Iterations

Communication costs pose a challenge in FL: to train effective global models, one might need to perform several federated learning iterations (or federated learning cycles). Each federated learning iteration reduces the efficiency of the whole federated learning process, as it requires an additional round of communication. Thus, it is important to analyse how different DP techniques influence the number of iterations required for the global model to converge. The privacy loss parameter ϵ for local models in different FL iterations is calculated based on the sequential composition theorem described in Section IV-D.

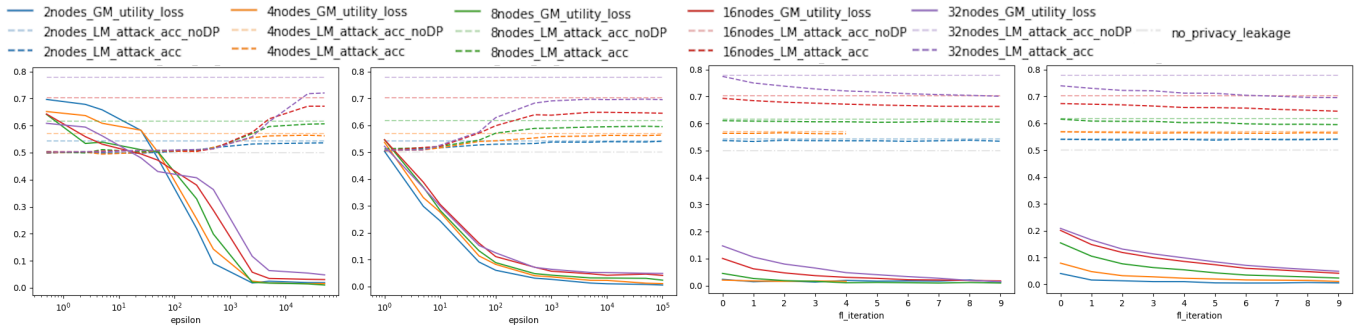
Considering the different numbers of nodes in FL without DP protection, we find that the global model converges faster (within the first two FL iterations) in the settings with two, four, and eight nodes. With 16 and 32 nodes, more than ten FL iterations are needed for the global model to converge. Therefore, we expect similar trends when applying DP and we use more FL iteration in the settings with more nodes.

Focusing on the number of FL iterations, we find that applying **output DP results in global models with high accuracy already after the first FL iteration** for both Loan and Purchase datasets (see Figures 1a and 1c), while DP-SGD requires more FL iterations to train a better global model. On the Loan dataset, in terms of privacy leakage, DP-SGD performs very similarly after the first and fifth FL iterations (Figure 1b bottom). At the same time, after the fifth iteration, the global model achieves a better accuracy score. In terms of privacy-utility trade-off, neither of the DP approaches seems



(a) Loan: OutpDP (b) Loan: DP-SGD (c) Purchase: OutDP (d) Purchase: DP-SGD

Fig. 1: Output DP and DP-SGD (2 nodes in FL) performance comparison based on global model accuracy and attack accuracy against local models, on different numbers of FL iterations. The colour indicates the point of testing (blue: results after the first federated iteration; orange: results after the fifth federated iteration); dotted lines indicate the baseline when no DP is applied.



(a) OutpDP (b) DP-SGD (c) OutpDP; $\epsilon = 10^4$ (d) DP-SGD; $\epsilon = 10^4$

Fig. 2: Output DP and DP-SGD performance on Loan dataset in FL settings with a different number of nodes. The global model's utility loss (GM_utility_loss) is computed as the difference between the global model's accuracy in FL without DP and with the corresponding DP approach. LM_attack_acc denotes the mean attack accuracy of all local models in the FL setting.

to perform better than the other on the Loan dataset in FL with two nodes.

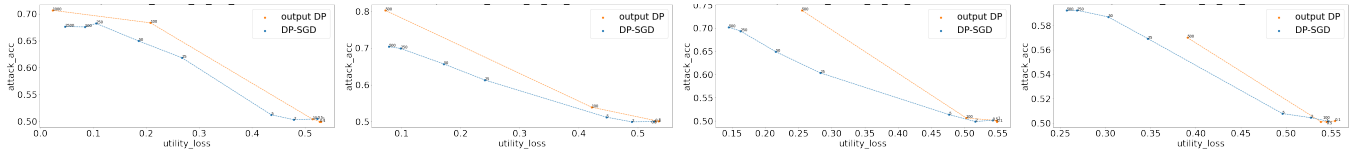
Analysing DP-SGD on the Purchase dataset (Figure 1d), we observe that training DP-SGD with one FL iteration results in a global model accuracy at the best case 10% lower than the baseline. At the same time, this does not bring any privacy gains, as in that case the ϵ is very high. When we train for more federated iterations, we achieve a better global model accuracy, which is in the best case 5% worse than the baseline. Nevertheless, in this case, the attack accuracy stays the same as the baseline without DP; thus, in order to gain more privacy in DP-SGD, one needs to sacrifice a large amount of the models' utility. On the Purchase dataset, we, therefore, note that output DP achieves a better trade-off between privacy and utility. Already at the first FL iteration output DP allows achieving global model accuracy only 3% lower than the baseline and at the same time reducing local models attack accuracy by 3%.

Comparing leakage from the models in the first and the last iteration of federated training, we see that models at the last FL iteration leak less data. The attack accuracy on local models is decreasing with each FL iteration, which can be explained by averaging the models in federated learning, better generalisation of local models and thus less overfitting to specific instances from the training set. That effect is especially pronounced on the Purchase dataset for both Output DP and DP-SGD (see Figures 1c and 1d). Therefore one should consider **using lower ϵ in the first FL iterations, as local models trained before the first aggregation are the most vulnerable to membership inference.**

B. Number of Nodes in Federated Learning

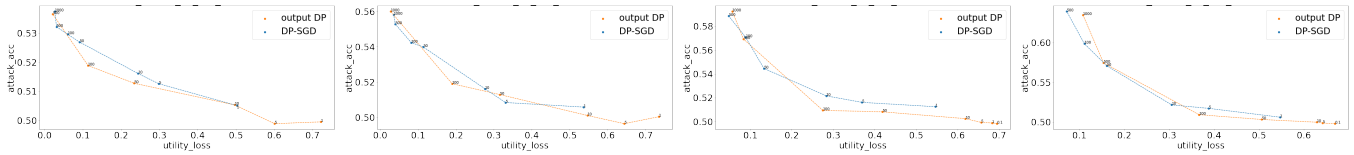
In this section, we consider the case when the training set is split with equal size between the clients: e.g. in the case of eight clients, each client has $10K/8$ samples for local training. Membership inference attack performs better on the local models when there is less training data at each client, as it is easier for local models to overfit the training data, which in turn benefits the MIA success rate. Therefore, we observe higher membership inference accuracy in the setting with more clients in FL without DP.

Utility loss in Figure 2 stands for the difference between the accuracy of the global model in FL without DP and the accuracy of the global model with DP (with the corresponding number of nodes in FL). For both output DP and DP-SGD, with a larger amount of nodes in FL, it is more difficult to achieve an accuracy comparable to FL without DP. With two, four and eight nodes, we still achieve a global model accuracy



(a) Two nodes (b) Four nodes (c) Eight nodes (d) 16 nodes

Fig. 3: Output DP and DP-SGD on Purchase dataset with different ϵ values (depicted on the plots near the data points) in Federated Learning with different numbers of nodes in the setting.



(a) Two nodes (b) Four nodes (c) Eight nodes (d) 16 nodes

Fig. 4: Output DP and DP-SGD with different ϵ values (depicted on the plots near the data points) on the Loan dataset in Federated Learning with different numbers of nodes in the setting.

similar to FL without DP (i.e. with a utility loss close to zero). At the same time, we observe that output DP provides a better trade-off between privacy and utility, allowing us to achieve low utility loss, while reducing the attack accuracy (see Figure 2a). **For a larger number of nodes in FL, both considered DP approaches result in higher utility loss**. They are comparable to each other in magnitude, losing at most 7% of the accuracy, but at the same time significantly reducing privacy risks. The attack accuracy is 10% lower for the setting with 32 nodes and 5% lower for the setting with 16 nodes when using DP-SGD (see Figure 2b). With output DP and 32 nodes in FL, the attack accuracy on the local models is almost 15% lower than FL without DP (see Figure 2a). With 16 nodes, we lose a bit less in utility (around 5%), and reduce the attack accuracy by 7%.

From Figures 2c and 2d, one can observe that with a larger number of nodes in FL, it takes more iterations for the model to converge to a lower utility loss. MIA accuracy decreases as well with utility loss. As mentioned above, this effect can be explained by the fact that local models generalise better after each FL iteration, due to the federated averaging. We showed that regardless of the number of nodes in FL, one should use a lower ϵ in the first few iterations to reduce the risk of inference from the local models.

C. DP-SGD versus Output DP

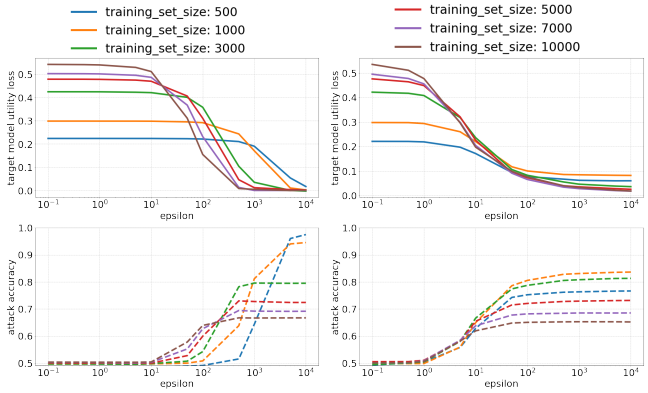
Figures 3 and 4 show the utility loss against the attack accuracy for models trained with different DP approaches and different ϵ. Both Figures 3 and 4 demonstrate how higher ϵ leads to higher privacy risks, confirmed by the attack accuracy. With an $\epsilon <= 5$, the attack accuracy for DP-SGD is close to a random guessing baseline, which implies that the model is immune to membership inference. At the same time, utility loss increases to 55%, which makes the global model useless for its actual classification task. One can see that for the Purchase dataset **DP-SGD provides a better privacy-utility trade-off than output perturbation for FL settings with all considered different number of nodes** (see Figure 3).

For the Location dataset, however, output perturbation outperforms DP-SGD in the settings with two (Figure 4a) and four (Figure 4b) nodes: e.g. in the setting with two nodes (Figure 4a) and an attack accuracy less than 52%, output perturbation with result in only 10% utility loss, while DP-SGD will achieve the same privacy level only with a cost of 25% utility loss. In FL scenarios with more modes (see Figure 4c, 4d), both considered DP approaches provide a very similar trade-off between privacy and utility. Nevertheless, **the usage of output perturbation can be preferable in cases when DP-SGD and output perturbation result in a similar privacy-utility trade-off, as output perturbation is computationally more efficient and requires fewer FL iterations to achieve an effective global model.**

D. Quantity skew in Federated Learning

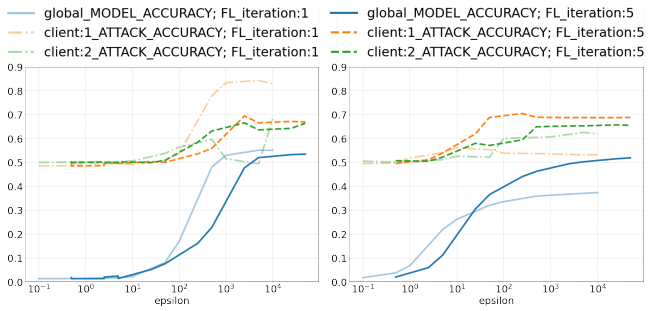
In a real-world federated setting, different clients often have data of different quantities and distributions. Non-IID data poses challenges for global model convergence in federated learning [26]. Another challenge that such data distribution entails is uneven privacy risks for different clients. Here, we consider the case of data quantity skew in federated learning, i.e. the data is unequally distributed among the clients. To investigate how DP will perform in such settings, first, we consider models trained on a different number of training samples in a centralised setting.

Figure 5 shows how output DP and DP-SGD perform on the models trained on datasets of different sizes, from 500 samples to 10K samples. One can observe that generally, membership inference attack accuracy is lower when attacking the models trained on a larger training set. That can be explained by the fact that models can remember and overfit training data better when they are trained with fewer samples, and, therefore, MIA works better on such models [27]. Interestingly, even with



(a) Output DP (b) DP-SGD

Fig. 5: Output DP and DP-SGD performance in centralised settings on Purchase dataset. Models trained with a different number of training samples (training_set_size). Target model utility loss denotes the difference between the accuracy of the target models trained without DP and the accuracy of the target models trained with DP.



(a) Output DP (b) DP-SGD

Fig. 6: Output DP and DP-SGD performance in FL with two nodes (client:1 and client:2) on Purchase dataset and skewed data distribution among the clients. Faded lines correspond to the first FL iteration, bright lines correspond to the results after the fifth FL iteration.

a very high ϵ, DP-SGD mitigates privacy risks better than output DP for the models which had smaller training sets. When the training set size is set to 1,000 samples (orange line), the attack accuracy goes up to 95% when using output DP (Figure 5a) and 85% when using DP-SGD (Figure 5b). When the number of training samples is more than 1,000, both approaches achieve quite similar results and show that privacy comes at a large utility cost. That leads us to the conclusion that **clients who train their models on small datasets should use DP-SGD rather than output DP**, as the first one allows for a better privacy-utility trade-off in that case.

To simulate quantity skew in federated learning we consider the case with two clients in FL and assign to the first client 25% of the data, while the second receives the remaining 75%; thus, client:1 has 2,500 samples, and client:2 has 7,500 samples. Figure 6 shows the corresponding FL training after the first and fifth FL iterations. When we apply output perturbation to achieve DP, we observe that client:1 has higher risks of privacy leakage, as it has relatively few samples compared to client:2 (see Figure 6a). We also notice that on the first FL iteration, the risk of privacy leakage for the client:1 is very high, and increases to 85% when using high ϵ. At the same time, on the fifth iteration, the attack accuracy on the client:1 local model is only around 70% with an $\epsilon > 10^3$. For the client:2 the results are the opposite: at the first FL iteration, the local model leaks less data than at the fifth FL iteration.

In Figure 6b), we observe, that DP-SGD manages to mitigate privacy risks for the client:1 at the first iteration even with a very high ϵ: the attack accuracy is close to the 50% (random guessing baseline). However, on the fifth FL iteration, the leakage from the client:1's local model increases up to 70%. On the fifth iteration, *both DP-SGD and output DP suggest a similar trade-off between privacy and utility*. This analysis of DP in FL with data distribution skew shows that different clients are affected differently by both DP-SGD and output DP. In future work, we aim to extend the current analysis to more non-IID settings in FL with DP and consider settings combining both DP approaches at different stages of the training.

VII. Conclusion and Future Work

In this paper, we conducted a comprehensive analysis of DP through output perturbation and DP-SGD in various federated learning settings. We considered the performance of these two approaches in terms of the utility of the global model and local models' privacy. We measured the empirical privacy risks via a membership inference attack, attacking both local and global models. We considered settings with different numbers of nodes in federated learning and also analysed the effect of different numbers of federated learning iterations.

From the experimental evaluation we can draw the following main findings:

- When applying output DP in FL, one can use fewer FL iterations to reach an optimal global model. **Output DP is thus more communication efficient than DP-SGD**. **Output perturbation is also more computationally efficient than DP-SGD**, as DP-SGD requires an increasing number of local iterations during gradient optimisation. Output DP is a computationally "cheap" privacy, as the noise is added only once after the model has been trained, while in DP-SGD the noise has to be added to the gradient after each batch.
- In FL settings with more than eight nodes, **both output perturbation and DP-SGD have a larger impact on the privacy and utility of the models and result in higher utility loss**.
- The privacy loss **parameter ϵ results in different levels of leakage for the different DP approaches** – and even for the same approach on different datasets or in different settings. Therefore, to find the best trade-off between privacy and utility, one has to investigate how different

ϵ influence the performance of the models and inference attacks for the particular case, dataset and DP approach.

- **DP-SGD suggests a better trade-off between privacy and utility** compared to output perturbation in most of the considered settings. However, in some settings the privacy-utility trade-off achieved by output perturbation and DP-SGD is similar, and due to higher efficiency, the output perturbation would be preferable to use. One of the main issues with output perturbation, however, is the limitation of the machine learning models to which output DP can be applied, as it requires deriving the sensitivity of the algorithm.
- Finding a good trade-off is a difficult task, as **DP has a potentially large impact on the model quality**. In some settings, neither of the considered DP approaches allowed for improved privacy without a (too) substantial reduction of the global model utility.
- In non-IID settings, FL **clients with smaller training sets are more prone to data leakage through the local models**, especially on the first FL iteration.

In future work, we aim to extend our evaluation of non-IID data scenarios in federated learning and consider different noise-adding strategies (e.g. more noise in the first iterations) to optimise the privacy-utility trade-off when using DP. We plan to consider other machine learning algorithms with known sensitivity bounds to combine them with output perturbation, e.g. SVM [28].

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics*, 2016.

[2] J. Zhang, H. Zhu, F. Wang, J. Zhao, Q. Xu, H. Li, and Z. Wang, "Security and privacy threats to federated learning: Issues, methods, and challenges," *Sec. and Commun. Netw.*, vol. 2022, jan 2022.

[3] A. Pustozerova and R. Mayer, "Information leaks in federated learning," in *In Proceedings of the Workshop on Decentralized IoT Systems and Security.*, 2020.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2017, pp. 3–18.

[5] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1322–1333.

[6] L. Zhu, Z. Liu, , and S. Han, "Deep leakage from gradients," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[7] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 308–318.

[8] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. 29, pp. 1069–1109, 2011.

[9] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, ser. STOC '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 169–178.

[10] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, 2021.

[11] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 639–648.

[12] N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. McMahan, S. Vassilvitskii, S. Chien, and A. Thakurta, "How to dp-fy ml: A practical guide to machine learning with differential privacy," 03 2023.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, p. 211–407, aug 2014.

[15] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2008.

[16] S. Song, K. Chaudhuri, and A. Sarwate, "Stochastic gradient descent with differentially private updates," in *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, ser. 2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings, 2013, pp. 245–248, 2013 1st IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 ; Conference date: 03-12-2013 Through 05-12-2013.

[17] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What Can We Learn Privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, Jan. 2011. [Online]. Available: http://epubs.siam.org/doi/10.1137/090756090

[18] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, ser. EdgeSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 61–66.

[19] L. Sun, J. Qian, and X. Chen, "Ldp-fl: Practical private aggregation in federated learning with local differential privacy," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 1571–1578, main Track.

[20] P. Kairouz, Z. Liu, and T. Steinke, "The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation," Feb. 2021. [Online]. Available: https://www.semanticscholar.org/paper/The-Distributed-Discrete-Gaussian-Mechanism-for-Kairouz-Liu/23099e2bf6e6675caf021fd1337e0988f8ed7d40

[21] I. Jarin and B. Eshete, "Pricure: Privacy-preserving collaborative inference in a multi-party setting," *Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics*, 2021.

[22] B. Avent, A. Korolova, D. Zeber, T. Hovden, and B. Livshits, "{BLENDER}: Enabling Local Search with a Hybrid Differential Privacy Model," 2017, pp. 747–764. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/avent

[23] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning Differentially Private Recurrent Language Models," Feb. 2018. [Online]. Available: https://openreview.net/forum?id=BJ0hF1Z0b

[24] I. Jarin and B. Eshete, "Dp-util: Comprehensive utility analysis of differential privacy in machine learning," in *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 41–52.

[25] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proc. VLDB Endow.*, vol. 5, no. 11, p. 1364–1375, jul 2012.

[26] A. Pustozerova, A. Rauber, and R. Mayer, "Training effective neural networks on structured data with federated learning," in *Advanced Information Networking and Applications*, L. Barolli, I. Woungang, and T. Enokido, Eds. Cham: Springer International Publishing, 2021, pp. 394–406.

[27] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282, 2017.

[28] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for svm learning," *Journal of Privacy and Confidentiality*, vol. 4, no. 1, Jul. 2012.