Analysing Utility Loss in Federated Learning with Differential Privacy

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

Jan Baumbach
Institute for Computational Systems Biology
University of Hamburg, Germany
jan.baumbach@uni-hamburg.de

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

***Abstract*—Federated learning provides the solution when multiple parties want to collaboratively train a machine learning model without directly sharing sensitive data. In Federated Learning, each party trains a machine learning model locally on its private data and sends only the models' weights or updates (gradients) to an aggregator, which averages locally trained models into a new global model with higher effectiveness. However, the machine learning models, which have to be shared during the federated learning process, can still leak sensitive information about their training data through e.g. membership inference attacks. Differential Privacy (DP) can mitigate privacy risks in federated learning by introducing noise into machine learning models. In this work, we consider two approaches for achieving Differential Privacy in federated learning: (i) output perturbation of the trained machine learning models and (ii) a differentially-private form of stochastic gradient descent (DP-SGD). We perform an extensive analysis of these two approaches in several federated settings and compare their performance in terms of model utility and achieved privacy. We observe that DP-SGD allows for a better trade-off between privacy and utility.**

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

I. Introduction

Machine Learning (ML) requires large amounts of data to train effective models. Data are often distributed and needs to be aggregated at a centralised place before training, e.g. from different mobile devices or medical institutions. However, sharing sensitive data can be impossible due to various regulatory constraints (e.g. with medical data), or in other cases, data owners might not be willing to share their private data with other parties. Federated learning (FL) proposes a new paradigm for training machine learning models: data owners, then called FL clients or nodes, can keep their sensitive data on-site, and share only machine learning models (parameters or gradients) trained on that data. A federated learning aggregator then collects all the local updates and averages them into a global model, which can be used for inference.

Federated Learning is already actively used for processing e.g. medical data [1], where data privacy is one of the key concerns. While FL eliminates the need to share sensitive data, it still requires the exchange of the models trained on that data. These models could be subject to inference attacks [2] (e.g. membership inference [3], model inversion [4], etc.), which allow adversaries to infer sensitive information about training data having access only to a machine learning model. To protect models from leaking sensitive information, one can use cryptographic approaches like Homomorphic Encryption (HE) [5] or Secure Multi-Party Computation (SMPC) [6]. These approaches, however, tend to have high computational and communication costs, especially with a large number of nodes in FL. Moreover, while securing locally trained models from inference, neither of these approaches can protect the final global model from inference attacks.

Differential Privacy (DP) [7] is a mathematical definition of privacy, which allows to set a bound on disclosure. We consider Differential Privacy as a mitigation strategy in FL, as this technique can mitigate privacy risks caused by different types of attackers having different access to the models in FL. In machine learning, DP can be achieved by introducing noise to the data, the model, or the model's output. Due to properties like composition and invariance under post-processing, DP can guarantee privacy for local and global models in federated learning. Therefore, DP mitigates privacy risks in FL coming not only from a malicious server, having access to the local model, but also from the malicious users of intermediate and final global models. Another benefit of Differential Privacy is the ability to calculate and regulate privacy loss by a privacy budget parameter - ϵ. The main challenge in the DP application is that the noise introduced to the model or data to protect their privacy inevitably decreases the utility of the models.

In this work, we focus on applying Differential Privacy in federated learning to mitigate privacy risks and investigate the privacy-utility trade-off provided by different DP techniques. We consider two approaches that allow achieving DP in FL to protect local and global models: *output perturbation*, where the noise is added to the trained model, and *DP-SGD*, where the noise is added during training. We investigate which technique results in a better trade-off between the model's privacy and utility.

This work provides a better understanding of the trade-off between the utility and privacy of the models in federated learning when using different DP approaches (output perturbation and DP-SGD). We present a comprehensive experimental analysis of output DP and DP-SGD and compare these two approaches in different FL settings.

The main findings of our work are as follows:

- DP-SGD results is a better trade-off between privacy and utility. With the same privacy budget parameter, DP-SGD causes lower utility loss than output perturbation, and therefore, it is preferable to use DP-SGD.

- For both DP-SGD and output perturbation, we find that training with a single FL iteration (one communication round) comes at a lower utility loss than training with more FL iterations. In the case of a higher epsilon (corresponding to lower privacy), however, training with more iterations was shown to be more effective for DP-SGD.
- Privacy through DP comes at a large utility cost: in the settings with lower epsilon (corresponding to a higher privacy level), the utility of the model drops significantly for both considered approaches.

The remainder of the paper is organised as follows. In Section II, we discuss existing works on DP in machine learning and federated learning. Section III provides the definitions and foundations of the approaches used in the evaluation. For reproducibility purposes, we provide a thorough description of our experimental setup in Section IV. The main findings from the experimental evaluation are described in Section V. In Section VI, we summarise the main contribution and future work.

II. Related work

Differential Privacy (DP) is widely used in privacy-preserving machine learning to analyse and mitigate privacy risks. In [7], Dwork et al. introduce differential privacy to mitigate privacy risks when querying statistics about the data. In subsequent work, they provide proofs for critical properties of DP, like sequential and parallel composition and invariance under post-processing [8].

In machine learning, DP can be applied at different stages and achieved by adding noise to the input data, objective function, gradients or trained model's weights. Input perturbation results in a higher utility loss than other methods [9]. Objective perturbation [10] can be applied only to machine learning models with a convex loss function. Output perturbation [11] limitation is that it requires knowing the sensitivity of a model (see Section III), however, it is an efficient way to introduce DP in federated learning, as it requires adding noise only once to the trained model.

One of the most popular approaches for achieving DP in ML was introduced in [12]. The authors presented a differentially-private version of the stochastic gradient descent optimisation algorithm (DP-SGD). The approach was widely adopted, as it is possible to use it with any ML model utilising SGD for optimisation, including neural networks.

There are several works considering DP in federated learning. Geyer et al. [13] focus on securing clients' contributions to the global model. They perturb the client's updates so the aggregator cannot tell where the updates came from. Other works consider hybrid approaches to mitigate the risks of inference from the local models in FL, e.g. combining SMPC and DP [14], [15], [16]. Jarin et al. [15] use SMPC to secure local models and add DP noise to the global model to secure it from the inference attacks of malicious clients. Adnan et al. [17] analysed the performance of DP-SGD in federated learning with IID (Independent and Identically Distributed) and non-IID data. Naseri [18] test DP against backdoor attacks in federated learning. They research scenarios when DP is used to protect local models from inference (local DP) and cases when DP protects only global models (central DP). They achieve local DP training local models with DP-SGD and central DP by perturbing the aggregating function at a server. Sun et al. [19] consider federated learning with neural networks and propose a mechanism based on DP to add noise to the weights of a neural network. Truex et al. [20] also consider DP in federated learning with neural networks and suggest a novel approach allowing clients in FL training complex models, such that each client is preserved from inference attacks. The approach is based on two steps: perturbation of complex models' parameters and selective sharing of these parameters at different FL iterations.

Most of the existing works considering DP in a federated setting utilise only the DP-SGD algorithm. From the works focusing on Differential Privacy applications in centralised machine learning, Jarin et al. [9] provide an analysis of Differential Privacy considering output and gradient perturbation. We extend their work and bridge the gap to federated learning, by comparing output perturbation and DP-SGD in federated learning settings. We conduct an extensive experimental evaluation and analyse which technique results in a better trade-off between privacy and utility when used in federated learning.

III. Differential Privacy

Definition 1 [7]: A randomised algorithm M with domain $N^{|x|}$ is (ϵ, δ)-differentially private if for all $S \subseteq Range(M)$ and for all $x, y \in N^{|x|}$ such that $||x-y||_1 \leq 1$:

$$Pr[M(x) \in S] \leq exp(\epsilon) Pr[M(y) \in S] + \delta$$

If $\delta = 0$, algorithm M achieves (ϵ)-DP. The main idea behind differential privacy is that randomised algorithms should behave similarly on inputs which differ only in one element. The privacy budget parameter $\epsilon \in (0, \infty)$ allows to regulate the privacy loss. Essentially, it denotes the similarity of the probability of the output for two functions that are queried on input datasets that differ only in one element. The higher the ϵ, the higher the privacy leakage. Parameter δ is a relaxation parameter and should satisfy the condition: $0 \leq \delta \leq 1$.

In [7], Dwork et al. showed that the privacy of the database can be preserved by adding noise according to the *sensitivity* of the function f. Formally [7], the sensitivity of a function $f : D \to \mathbb{R}$ is defined as the smallest number $S(f)$ such that for all $x, x' \in D$ which differ in a single entry,

$$||f(x) - f(x')||_1 \leq S(f)$$

The composition property and the immunity to postprocessing are crucial properties of differential privacy. **Sequential composition** guarantees that the output of the application of multiple DP mechanisms on the same database still is differentially private, and their cumulative privacy can be calculated as a sum of individual privacy losses [8]. In our work, we use sequential composition to calculate the privacy loss for the local models, which are trained on the same dataset for several FL iterations. Considering that each client has the same ϵ at each FL iteration, the privacy loss of the local model on the $n-$th FL iteration will be $n * \epsilon$. **Parallel composition** [21] guarantees that if n (ϵ_i, δ_i)-DP mechanisms $(i \in [1, n])$ are applied on n disjoint datasets, the composition of these mechanisms will be $(max_{i\in[1,n]}(\epsilon_i), max_{i\in[1,n]}(\delta_i))$-DP.

A. Differential Privacy via Output Perturbation

Output perturbation (short: Output DP) was first used by Chaudhuri and Monteleoni [11] when they applied the sensitivity method from Dwork et al. [7] on the Logistic Regression model. They found that the sensitivity of a regularised Logistic Regression is at most $\frac{2}{n\lambda}$, where λ is a regularisation parameter. Knowing the sensitivity, one can calculate the noise needed to achieve differential privacy. In our work, we use the Gaussian mechanism [8] to achieve (ϵ, δ)-differential privacy. The noise is drawn from Gaussian distribution $N(0, \sigma^2)$, where $\sigma = S(f; 2)\sqrt{2ln(1.25/\delta)}/\epsilon$ [8] and $S(f; 2)$ is the l_2-sensitivity of the model. To get a differentially private model from a trained machine learning model, we just add the noise drawn from the Gaussian distribution to the models' weights: $\theta_{dp} = \theta + noise$.

B. Differentially-Private Stochastic Gradient Decent

One of the most popular approaches for achieving DP in machine learning is the Differentially-Private Stochastic Gradient Decent (DP-SGD) [12]. The idea is to add noise to the gradients used to update the model parameters during the model's training. DP-SGD algorithm achieves differential privacy by randomly sampling a batch from the training data, computing the gradient for each sample, and then clipping (bounding) each gradient. After that, noise drawn from the Gaussian distribution is added to the sum of the gradients. Lastly, one can update the model with the now "noisy" gradients.

IV. Experimental setup

In the following, we describe the setup for our experimental evaluation in detail. To enable reuse and reproducibility, our source code is publicly available[1].

A. Datasets

In our evaluation, we use two datasets from previous works on DP:

The **Purchase-100** (Purchase) dataset consists of 600 binary attributes representing different products that individuals can buy. The classification task has 100 classes, which represent groups of customers with different purchase behaviour. We use a preprocessed version of the dataset[2], and to be able to compare to the baseline results from [3] we as well use a subset of 10K samples to train the model. For centralised learning with Logistic Regression, we achieve an accuracy score of 56%, using a learning rate of 0.001, l_2-regularisation of $1e-4$ and 50 iterations.

The **LendingClub-Loan** (Loan) dataset was used by [9], and is available on Kaggle[3]. The data represents individuals who want to get a loan and contains information about them, like education, salary and others. The classification task is to classify individuals in one of the six risk groups. We use the code from [9] to preprocess the data in the same way. As with the Purchase dataset, we also randomly selected 10K samples for training. We achieve an accuracy score in the centralised setting of 86%, with a learning rate of 0.01, l2-regularisation of $1e-6$ and 200 iterations.

B. Differential Privacy

We use grid search in a centralised setting to find optimal hyperparameters, first without DP and later when we apply differential privacy in the form of output DP or DP-SGD. While using DP-SGD, we had to increase the number of iterations (from 50 to 200) to achieve an accuracy score close to the centralised setting even with a very high epsilon. We found the following optimal parameters for DP-SGD: for the Loan dataset with a mini-batch size of 20 and a norm bound of 10, and for the Purchase dataset, the same mini-batch size of 20 and norm bound of 2. For output DP, we found the same optimal parameter for l2-regularisation, namely $1e-4$, for both datasets. For the parameter δ, as recommended in the literature [22], we use $\delta << 1/n$, which is in the case with both datasets $\delta = 1e-5$.

C. Federated Learning Setup

In federated learning, we consider settings with two, four, eight, 16 and 32 nodes. We split the training set randomly and equally among the nodes. We use the federated averaging algorithm [23] to compute the global models. We use the same hyperparameters for training the local models at each node.

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, which represents an upper bound of achievable effectiveness – though training a centralised model is often not possible due to e.g. data protection regulations. Still, in the evaluation of the experiment, we compare the effectiveness of the global and local models in federated learning to the centralised baseline presented in Section IV-A.

During the evaluation, we especially focus on the utility of the global and local models, measured by the accuracy score. The *utility loss* caused by DP is estimated by the formula:

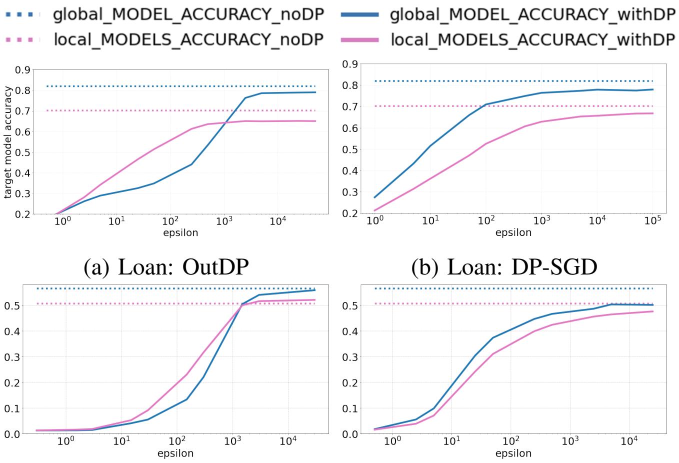
$$1 - \frac{Model\ Accuraccy\ with\ DP}{Model\ Accuraccy\ without\ DP}$$

The relation between the accuracy score and ϵ, which represents privacy loss, is used to evaluate the privacy-utility trade-off.

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

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

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



(a) Loan: OutDP

(b) Loan: DP-SGD

(c) Purchase: OutDP

(d) Purchase: DP-SGD

Fig. 1: Local and global models accuracy in federated learning: Output DP in FL with 16 nodes (a, c), DP-SGD in FL with four nodes (b, d). The mean accuracy of the local models is indicated by pink, and global model accuracy by blue lines. Dotted lines indicate accuracy w/o DP, solid lines with DP.

V. Results and Discussion

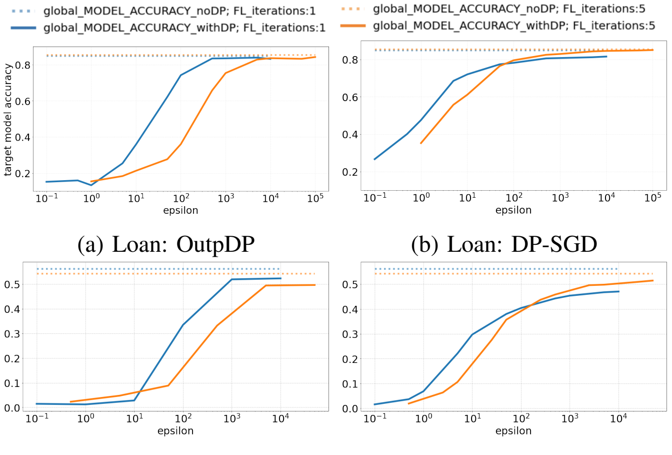
In this section, we present and discuss the results of our experimental evaluation. We start by comparing the utility and privacy leakage of the global and local models in FL trained with different DP approaches. We then discuss the federated learning iterations, which have an impact on the efficiency, as well as the effect of having different numbers of nodes in the federation. We then compare specifically the DP-SGD and the Output DP to determine which of the approaches provides a better privacy-utility trade-off.

A. *Global vs. Local Models*

The ideal benchmark for federated learning is that the collaboratively trained machine learning model achieves effectiveness close to a (hypothetical) centralised setting. This has to be seen as an upper bound of achievable accuracy, as the centralised setting is often not a viable alternative, e.g. when data sharing and centralising are not feasible or allowed due to regulations. However, even in such scenarios, federated learning can still be beneficial if the effectiveness of the global model is higher than the effectiveness of local models trained by federated learning clients on their local data [24]. Therefore, we start our analysis by investigating local and global models' performance (see Figure 1). We evaluate to which extent DP influences the global model accuracy and which privacy-utility trade-off it provides.

Comparing DP-SGD and OutputDP performance on the Loan dataset (see Figures 1a and 1b), we observe that with relatively high ϵ ($10^3 < \epsilon < 10^4$), the global model achieves higher accuracy score when we use output DP. However, in a privacy-sensitive setting with $\epsilon \leq 10$, DP-SGD provides a better privacy-utility trade-off. The accuracy, in that case, however, drops substantially for both DP-SGD (the accuracy is around 40% lower than the non-DP global model) and for output DP (50% lower than the non-DP global model).

A similar effect can be observed on the Purchase dataset (see Figure 1c): when performing output perturbation on local models with $\epsilon > 10^3$, the global model achieves an accuracy close to the global model without DP. DP-SGD does not allow achieving high accuracy even with a very high ϵ. However, with the lower ϵ, DP-SGD results in lower utility loss than output DP Figure 1d): for $\epsilon = 10$, DP-SGD achieves accuracy of a global model close to 20%, while output DP results in the accuracy of only 5%.



(a) Loan: OutpDP

(b) Loan: DP-SGD

(c) Purchase: OutDP

(d) Purchase: DP-SGD

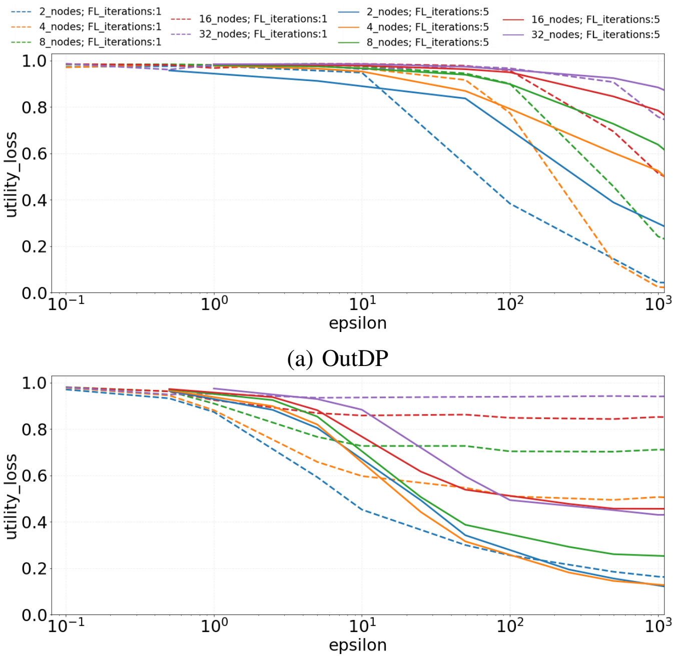
Fig. 2: Output DP and DP-SGD in FL with two nodes, evolution of the global model accuracy throughout over several FL iterations. The colour indicates the point of testing (blue results after the first, orange after the fifth federated iteration); dotted lines indicate the baseline when no DP is applied.

B. *Federated Learning Communication Rounds*

In this section, we investigate in detail the efficiency of DP-SGD and output DP in terms of communication rounds (or FL iterations). The goal would be to reduce the number of FL iterations to avoid unnecessary communication and possible failures. We evaluate models in terms of utility against ϵ to estimate the privacy-utility trade-off.

From Figure 2, we observe that training models with one FL iteration (or one communication round) results in a better privacy-utility trade-off. This trend is more pronounced for output perturbation on both considered datasets Figures 2a and 2c). With a privacy budget of $\epsilon = 10^2$, the accuracy of the global models trained with only one iteration is 10% lower than the accuracy of the non-DP global model for the Loan dataset and 20% lower for the Purchase dataset, while after five FL iterations with the same privacy budget, the accuracy drops by more than 40% for both datasets.

With DP-SGD (see Figures 2b and 2d) in a not strict privacy setting ($\epsilon \geq 10^2$) training with more FL iteration results in higher utility. However, with $\epsilon \leq 10$, training with one



(a) OutDP

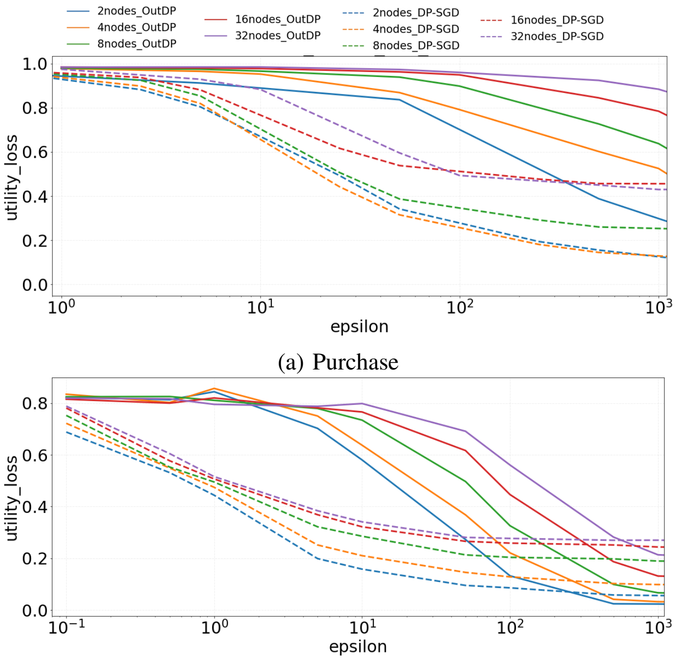
(b) DP-SGD

Fig. 3: Utility loss by Output DP and DP-SGD of global models on the Purchase dataset, with different number of nodes in FL, and after the first (dashed line) and fifth FL iteration (solid line).

iteration provides better accuracy for both datasets. Comparing DP-SGD and output DP performance we again observe that DP-SGD results in a better privacy-utility trade-off: for $\epsilon = 10$ DP-SGD results only in 10% accuracy loss on the Loan dataset, while output DP drops accuracy by more than 40% for the same ϵ. On the Purchase dataset with $\epsilon = 10$ the accuracy drops to 30% when using DP-SGD, while with output DP the accuracy drops to only 5%.

In Figure 3, we provide results for the scenarios with more nodes in the FL settings to determine if these trends stay consistent regardless of the number of nodes in FL. For output DP (see Figure 3a), we observe that in settings with higher privacy ($\epsilon \leq 10$), all the trained global models result almost in 100% utility loss, which makes the models unusable. For $\epsilon > 10$, training only with one FL iteration results in a better privacy-utility trade-off, meaning that for the fixed privacy budget, the utility loss is lower if the federated learning process continues only with one communication round.

With DP-SGD (see Figure 3b), we notice that training with one FL iteration results in more effective global models than training with more iterations, but only in the privacy-sensitive settings with a low privacy budget ($\epsilon < 10$). For a higher privacy budget, training models with more FL iterations results in substantially lower utility loss (in Figure 3b we show results with five FL iterations). The larger the number of nodes in the federation, the lower the utility loss becomes with every federated iteration. This is the case for each considered setting,



(a) Purchase

(b) Loan

Fig. 4: Output DP (solid lines) compared to DP-SGD (dashed lines) in FL with different number of nodes.

except the one with two nodes (blue lines), where training with one FL iteration allows training global models with lower utility loss for all considered ranges of ϵ.

C. *DP-SGD versus Output DP in Federated Learning*

In this section, we select the best results achieved using output DP and the best achieved results with DP-SGD, and compare them with each other in federated settings with different numbers of nodes on Purchase and Loan datasets. From Figure 4, we observe that for both datasets **DP-SGD results in a better privacy-utility trade-off than output DP**. The difference in the utility loss is especially pronounced for the settings with a lower privacy budget ($\epsilon < 10^2$).

On the Purchase dataset (see Figure 4a), with $\epsilon = 1$, both approaches result in almost total utility loss. With $\epsilon = 10$, utility loss from DP-SGD decreased to the range from 60% to 90% (depending on the number of nodes), while output DP stays at 90% to 100% utility loss. With $\epsilon = 10^2$, DP-SGD results in 20%-50% utility loss, while output DP results in 70%-90% loss in the utility. One can clearly see the advantage of using DP-SGD, instead of output DP, however, both approaches cause substantial utility loss of the global model.

Figure 4b shows the results on the Loan dataset. Similarly to the results on the Purchase dataset, DP-SGD achieves lower utility loss, than output DP for $\epsilon \leq 10^2$. For $\epsilon < 1$, the accuracy of the global model with output DP drops by 80%, while for $\epsilon = 1$, DP-SGD results in the range of 40%-50% utility loss for different numbers of nodes. The gap between output DP and DP-SGD results is even higher for $\epsilon = 10$:

DP-SGD causes utility loss in the range 15%-35% and output DP results in the range 60%-80%. Output DP outperforms DP-SGD in terms of utility in the low privacy setting with $\epsilon = 10^3$, however in such cases, the privacy of the models is close to the one of the models trained without Differential Privacy.

VI. Conclusion and Future work

Differential Privacy can be applied to mitigate residual privacy risks in federated learning, which stem from the local and global models leaking sensitive information about their training data. Differential Privacy in machine learning can be achieved, among others, through applications of output perturbation mechanism or DP-SGD. While increasing privacy, applying DP inevitably results in a utility loss. In this paper, we provided an extensive analysis of the effects of DP-SGD and output perturbation in different federated learning settings, with different numbers of nodes. We considered different numbers of FL iterations for training global models. We conclude that DP-SGD provides a better trade-off between the privacy and utility of the models.

In future work, we will focus on exploring the implications of non-IID data in federated learning, and its influence on the performance of Differential Privacy. Our future work will consider the evaluation of alternative DP mechanisms applicable in machine learning, such as objective perturbation and input perturbation. Additionally, we will investigate the effects of DP in the context of federated learning across various machine learning algorithms, including Support Vector Machines (SVM) and neural networks, to gain insights into its impact in diverse scenarios.

Acknowledgements

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. COMET is managed by FFG.

References

[1] R. Torkzadehmahani, R. Nasirigerdeh, D. B. Blumenthal, T. Kacprowski, M. List, J. Matschinske, J. Späth, N. K. Wenke, B. Bihari, T. Frisch, A. Hartebrodt, A.-C. Hausschild, D. Heider, A. Holzinger, W. Hötzendorfer, M. Kastelitz, R. Mayer, C. Nogales, A. Pustozerova, R. Röttger, H. H. H. W. Schmidt, A. Schwalber, C. Tschohl, A. Wohner, and J. Baumbach, "Privacy-preserving Artificial Intelligence Techniques in Biomedicine," Jun. 2022, arXiv:2007.11621 [cs].

[2] A. Pustozerova and R. Mayer, "Information leaks in federated learning," in *Workshop on Decentralized IoT Systems and Security.*, 2020.

[3] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *IEEE Symposium on Security and Privacy (SP)*, May 2017.

[4] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures," in *22nd ACM SIGSAC Conference on Computer and Communications Security (CCS).* ACM, 2015.

[5] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *41st Annual ACM Symposium on Theory of Computing (STOC).* ACM, 2009.

[6] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively Secure Multi-Party Computation," in *28th Annual ACM Symposium on Theory of Computing (STOC).* ACM, 1996.

[7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Theory of Cryptography*. Springer, 2006.

[8] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, Aug. 2014, place: Hanover, MA, USA Publisher: Now Publishers Inc.

[9] I. Jarin and B. Eshete, "DP-UTIL: Comprehensive Utility Analysis of Differential Privacy in Machine Learning," in *12th ACM Conference on Data and Application Security and Privacy (CODASPY).* ACM, 2022.

[10] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially Private Empirical Risk Minimization," *Journal of Machine Learning Research*, vol. 12, no. 29, 2011.

[11] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Advances in Neural Information Processing Systems*, vol. 21. Curran Associates, Inc., 2008.

[12] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *ACM SIGSAC Conference on Computer and Communications Security (CCS).* ACM, 2016.

[13] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," Mar. 2018, arXiv:1712.07557 [cs, stat].

[14] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A Hybrid Approach to Privacy-Preserving Federated Learning," in *12th ACM Workshop on Artificial Intelligence and Security (AISec).* ACM, 2019.

[15] I. Jarin and B. Eshete, "PRICURE: Privacy-Preserving Collaborative Inference in a Multi-Party Setting," *ACM Workshop on Security and Privacy Analytics*, 2021.

[16] J. Matschinske, J. Späth, M. Bakhtiari, N. Probul, M. M. Kazemi Majdabadi, R. Nasirigerdeh, R. Torkzadehmahani, A. Hartebrodt, B.-A. Orban, S.-J. Fejér, O. Zolotareva, S. Das, L. Baumbach, J. K. Pauling, O. Tomašević, B. Bihari, M. Bloice, N. C. Donner, W. Fdhila, T. Frisch, A.-C. Hauschild, D. Heider, A. Holzinger, W. Hötzendorfer, J. Hospes, T. Kacprowski, M. Kastelitz, M. List, R. Mayer, M. Moga, H. Müller, A. Pustozerova, R. Röttger, C. C. Saak, A. Saranti, H. H. H. W. Schmidt, C. Tschohl, N. K. Wenke, and J. Baumbach, "The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach," *Journal of Medical Internet Research*, vol. 25, p. e42621, Jul. 2023.

[17] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Scientific Reports*, vol. 12, no. 1, Feb. 2022.

[18] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and Central Differential Privacy for Robustness and Privacy in Federated Learning," in *Network and Distributed System Security Symposium (NDSS).* Internet Society, 2022.

[19] L. Sun, J. Qian, and X. Chen, "LDP-FL: Practical Private Aggregation in Federated Learning with Local Differential Privacy," in *30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[20] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "LDP-Fed: Federated Learning with Local Differential Privacy," in *3rd ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys).* ACM, 2020.

[21] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, ser. SIGMOD '09. New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 19–30.

[22] N. Ponomareva, S. Vassilvitskii, Z. Xu, B. McMahan, A. Kurakin, and C. Zhang, "How to DP-fy ML: A Practical Tutorial to Machine Learning with Differential Privacy," in *29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).* ACM, Aug. 2023.

[23] 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.

[24] A. Pustozerova, A. Rauber, and R. Mayer, "Training Effective Neural Networks on Structured Data with Federated Learning," in *Advanced Information Networking and Applications.* Springer, 2021, pp. 394–406.