Image Classification on edge simulation based on Federated Learning

LI JINYUAN and CHAI ZHENGHAO
Shanghai Jiao Tong University

The increasing utilization of edge devices leads to rich data in private devices that is suitable for model training to improve the user experience. For privacy concern, the data can not be retrieved directly from client devices. So federated learning is introduced to obtain the data with less privacy loss, where edge devices train the models locally with user dataset and the central server aggregates parameters from local models to decouple the model training from the need for the central server accessing the raw user data. We set up the federated structure on the image classification task and test its results. Considering the remaining privacy risk in federated learning that the information of certain client's data could be revealed through the trained model, we apply differential privacy preserving mechanism in federated optimization to improve the privacy protection result by hiding one client's contribution in model training and analyze the privacy loss quantitatively.

Categories and Subject Descriptors: [**Machine Learning**]: Distributed Deep Learning

General Terms: Federated Learning

Additional Key Words and Phrases: Edge computing, Image Classification, Differential Privacy

1. INTRODUCTION

Today cell phones have become widely used and thus cell phone user generate massive amount of useful data every day. The data can be potentially used to train models with labels inferred from user interaction in supervised tasks and make models more powerful to improve the user experience. For example, the photographs taken and classified by users can potentially train the more intelligent image classification models.

Though users hold large amount of useful data on their devices, there are several problems to retrieve the data for training models. One of the problems can be quantity of the data may be too large for directly transmitting thus data can not be logged to the data center directly for training.

A more troublesome problem is that considerable part of the data is privacy sensitive, which also prevents the data from directly usage in the central server. Thus, Federated Learning is introduced to address such problem.

And the security problem also requires that users' information cannot reveal through analyzing the trained model to prevent the attackers with full knowledge of the training mechanism and access to the model's parameters. For this reason, some encryption methods are required for privacy protection and we suggest the differential privacy mechanism into federated structure to analyze and address the problem.

2. BACKGROUND AND RELATED WORKS

2.1 Federated Learning

The idea of Federated Learning is introduced to overcome the problem of privacy leakage. In Federated Learning, model training is done by a central server and several user devices where the training data distributed on the user devices. The server maintains a global model and each device owns a local dataset generated from users' behavior and a local model which is received from the central server. Some devices train the local model using their local datasets and obtain the update of the local model. The central server aggregates all the updates from devices and generates the update to the global model. With Federated Learning, the cloud is kept from accessing user data and the updates are ephemeral and never contain more information than the raw training data, thus the privacy the security risks can be reduced compared to the centralized method.

Google directs the attention to the massive data generated by cell phone users and investigates the Federated Learning technique [Brendan McMahan Eider Moore Daniel Ramage Seth Hampson Blaise AgüeraAg et al. 2016] introducing the basic Federated Averaging Algorithm to deal with the federated optimization. They use Federated Learning to achieve the mobile keyboard prediction task [Hard et al. 2018] that represents one of the first application of federated language model in commercial setting offering privacy advantages.

2.2 Differential privacy

Differential privacy [Dwork et al. 2014] serves as a rigorous standard for privacy guarantees for algorithms on data analysis. In our experiments, for instance, we can view each training dataset as a set of image-label pairs. We define that two of these pairs are adjacent if they differ only in a single entry. We use the definition as follows:

A randomized mechanism $M : D \to R$ with domain D and range R satisfies (ϵ, δ) differential privacy if for any two adjacent inputs $d, d' \in D$ and for any subset of outputs $S \subset R$ it holds that

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

In this definition, δ accounts for the probability that plain ϵ differential privacy is broken. The composition theorems enable application in the case of learning algorithm: if all the components of a mechanism are differentially private, then so is their composition. The composition privacy implies degradation of privacy preservation if datasets contain correlated components, or, in our experiments, the training data contributed by the same individual is train by several rounds.

2.3 Noise addition mechanism

A common approach for approximating a deterministic real-valued function $f : D \to R$ with a differentially private mechanism is adding the noise calibrated to $f's$ sensitivity S_f, which is defined as the maximum of the absolute distance $|f(d) - f(d')|$ where d and d' are adjacent inputs. For example, the Gaussian noise mechanism is defined as:

$$M(d)f(d) + N(0, S_f^2\sigma^2)$$

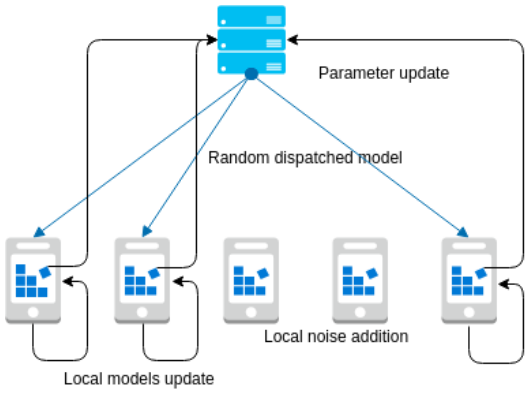

Fig. 1. Illustration of Dp SGD Algorithm

where $N(0, S_f^2\sigma^2)$ is the normal (Gaussian) distribution with mean 0 and standard deviation $S_f\sigma$. A single application of the Gaussian mechanism to function f of sensitivity S_f satisfies (ϵ, δ) differential privacy if $\delta \leq \frac{4}{5}\exp(-(\sigma\epsilon)^2/2)$

The differential private noise adding mechanism is specified by He et al [He et al. 2017]. and some mathematical foundations of the differential privacy preserving mechanism are established. Then Abadi et al.[Abadi et al. 2016] introduce the differential privacy mechanism into deep learning for privacy and security improvement in conventional neural networks training that provide a way to hide contribution of the data points in model training.

2.4 Dp SGD algorithm

Recently, [Abadi et al. 2016] proposed a differentially private stochastic gradient descent algorithm, which works similar to mini-batch gradient descent in the sense of distributed learning but the gradient averaging process is from the approximation by Gaussian model. What's more, mini-batches are mainly distributed by random sampling of data. In common approach, a privacy accountant keeps trac of δ and stops the round of training once the given threshold is reached. That means, training would stop once the system detects that the privacy loss for a client is used up, and further utilization of personal training data would become hazardous to one's privacy.

2.5 Moments accountant

Many research has studied the privacy loss condition for a particular noise distribution as well as the compositions of different privacy loss. As for the Gaussian noise condition, if we choose δ to be $\sqrt{2\log\frac{1.25}{\delta}}/\epsilon$, by standard arguments from [Dwork et al. 2014] each step is (ϵ, δ) differentially private with respect to the sub batch. We can ragard the sub batch as a random sample from the database. By the privacy amplification theorem [Dwork], we can know that each step is $(O(q\epsilon), q\epsilon)$ differentially private with the sample ratio $q = L/N$. Other works show that the best overall bound is the strong composition theorem.

2.5.1 *Privacy loss random variable.* The moment accountant is generally used to keep track of the bound of the privacy loss. It can help to provide much tighter bound of the privacy loss.

In [Abadi et al. 2016], privacy loss is a random variable dependent on the random noise added to the algorithm. It claims that a mechanism M is (ϵ, δ) differentially private is equivalent to a certain bound on $M's$ privacy loss random variable. Also, the tail bound is beneficial, while composing directly from it may result in a looser bound. Instead we can compute the log moments of the privacy loss. Combine the moment bounds with the standard Markov inequality to obtain the tail bound to compose the privacy loss in the sense of differential privacy loss.

To be more specific, for neighboring databases $d, d' \in D^n$, a mechanism M, auxiliary input aux, and an outcome $o \in R$, define the privacy loss at o as

$$c(o; M, aux, d, d') \log \frac{\Pr[M(aux, d) = o]}{\Pr[M(aux, d') = o]}$$

2.6 Conditions of differential privacy

A necessary and sufficient condition of ϵ differentially private is given in the following theorem.([He et al. 2017])

A is ϵ differentially private if and only if the following two conditions hold,

c1: zero measure of the zero-point set

$$\mu(\cup_{i=1}^{n}\Phi_i^0) = 0$$

where $\Phi_i^0 = \{z|f_{\theta_i}(z) = 0, z \in R\}$ is the zero point set and $\mu(\bullet)$ is the Lebsque measure.

c2: there exists a positive constant c_b such that

$$\sup_{\hat{\sigma}\in[-\sigma,\sigma], f_{\theta_i}(z)\neq 0} \frac{f_{\theta_i+\hat{\sigma}}(z)}{f_{\theta_i}(z)} \leq c_b$$

$\forall i \in V$, where the acnodes space (isolated point) of both $f_{\theta_i+\hat{\sigma}}$ and f_{θ_i} are not considered.

3. MOTIVATION

Recently, a lot of research has been conducted to deal with privacy problem in data mining. We specify this problem under the context of Federated learning, which is a novel idea and framework in the field of Edge computing and Deep learning.In Federated Learning, privacy protection is achieved by locally training the models in client devices and keeping the user data from the central server. However, this method may still have some risks of privacy leakage. The parameters of the local models from client devices may still reveal the information about user data used for training model. The method could be exposed to differential attacks, which could start from users' contributing during federated optimization and in such attack, a client's information can be revealed through analyzing the global model deliberately with another knowledge of the model mechanism. In fact, some other papers point out that when dealing with privacy problem in data processing and analysis, the rigorous approach should be the (δ, ϵ) representation under its mathematical definition.

Considering the defect of federated learning, the application of differential privacy preserving mechanism in federated optimization may be helpful[Geyer et al. 2017]. Using the client sided differential privacy preserving federated optimization, whether a certain client's data has contributed in model training may be hided to some extent when analyzing the learned model, thus a client's data can be protected from attacks by someone knowing the training mechanism and accessing to the model parameters. And the application of differential privacy can also provide a quantitative indicator for privacy analysis which Federated Learning do not focus on.

In the project, we set the background to the image classification model aiming as a photo classifier helping user classify the

photos they took. So we implement federated learning on the image classification task and compare the federated structure with the centralized structure. Then we implement the differential privacy preserving mechanism in federated optimization to improve the basic federated learning in privacy protection and analyze the privacy loss quantitatively to test the improvement of the results.

4. PROBLEM FORMULATION

4.1 Research objectives

We are looking forward to formulating this problem under the (δ, ϵ) representation and carrying out the deployment of this model. We mainly focus on the application field as image classification for an easier approach and confine the model under the topic of Federated learning, which means that we should obey the basic average framework of the target learning model. In order to preserve models' comparability, we just add further restriction on the traditional privacy-preserved learning model.

We mainly propose that those differential privacy mechanisms can also apply to the Federated learning models, especially under the noise-adding mechanism. We also simulate this model on the single-node platform and predict the output with the input of traditional MNIST dataset. The outcome shows that the noise-adding mechanism is a plausible approach of preserving client's privacy and the accuracy loss can be acceptable given client's total privacy budget.

4.2 Nature of the problem

Privacy preservation is a practical and enormous application field in Network Security and Data Processing, therefore, we mainly focus on the application issue in Federated learning topic, which can reduce much background knowledge requirement and save our energy.

In fact, Federated learning is a new topic just because it transfers the traditional model platform from Server to Edge devices. Such model mainly utilizes existent algorithms and models from the field of distributed learning, also bringing other edge-specified issues into the field, including communication cost and non-IID data. In order to do the research under the issue which is generic in both server and edge devices, we select the privacy preservation problem as our research target.

Practically speaking, introducing privacy concern into those existent learning models has so far caused only detrimental effects towards the final outcome, which means the relationship between privacy preservation and model's outcome is a trade-off. Introducing differential privacy into our model has another benefit, since the benchmark and criteria would differ a lot if different researchers choose different index, it is significant to discuss and compare the outcome under the same criterion. The goal for the researches in this field would mainly be bounding some parameters as much as possible to reduce its effect on overall performance.

5. PROPOSED METHODS

5.1 Federated Averaging Algorithm

We first implement the basic Federated Learning using Federated Averaging Algorithm, which adapts the idea of simple stochastic gradient descent into federated optimization problem, resolving the challenges including information overhead between clients and server, non-IID and unbalanced distributed data. The basic idea of the algorithm is random selection in the client device for computing efficiency and taking the weighted average of the local parameters. The algorithm is described as below.

—Assume there are K client device in total with a fixed local dataset. The global model in the central server owns the initialized parameter. ω_0

—In each communication around t, randomly select a fraction C of devices and send the current global model ω_t to them.

—Then each selected client device k performs training locally on its local dataset with SGD on their local model ω_t^k with E echoes and batch size of B to compute the updateω_{t+1}^k. In each echo, divide the dataset into batches and in each step, $\omega_t^k \leftarrow \omega_t^k - \eta \nabla l(\omega_t^k; b)$ where b is a batch of local dataset with size B, and η is the learning rate.

—Devices send the update ω_{t+1}^k to the server and the server aggregates the update by weighted average of the updates to update the global model $\omega_{t+1} \leftarrow \sum \frac{n_k}{n} \omega_{t+1}^k$, where n_k is the data size of each local dataset and $n = \sum n_k$.

—By calculating the privacy cost and evaluating the new central model, training is either stopped or the communication would start a new round.

Those basic laws show that clients never share data between each other, only central parameter server would coordinate model parameters with the clients.

5.2 Federated learning with differential privacy

In the traditional Federated learning framework, the central curator averages client models (weight parameters) at the end of each communication round. We apply the randomized mechanism to provide noise element into the communication message between clients and curator. This approach is effective for hiding the concrete single client's data distribution in both aggregation process and local learning process.

The specified random mechanism lies in following detailed steps:

—Random sub-batch sampling
If the total number of clients is N, in each communication round a random subset K_t of smaller size is sampled. The curator will distributes the central model w_t to only those random selected clients. The central model is trained and updated by the client's on their edge devices based on their own data. The selected clients in this round now hold their local models $\{w^k\}(k < |K|)$. The difference which is regarded as the update for certain clients are sent back to the central curator at the end of each communication round.

—Choose the clipping value S
GM is usually used to distort the sum of all updates from clients. We can force the dataset to have certain sensitivity by using scaled version instead of the unbounded true updates. we can let $\Delta \bar{w}_k = \Delta \frac{w_k}{\max(1, \frac{\|\Delta w_k\|_2}{S})}$. Here we use to make the second norm of weight bounded to S. Thus, the sensitivity of the scaled updates after the summing process is upper bounded by S. Now, our GM model can add noise which is scaled to sensitivity S to the scaled updates. There is trade-off in clipping noise contribution. S should be chosen small such that we could control the noise variance to be small. On the other hand, we want to maintain as much of the original contributions as possible. Therefore, we could let $S = median\{\Delta w^k\}_{k \in Z_t}$.

—Iterate σ and m

Given S, the ratio $r = \sigma^2/m$ will govern the distortion and privacy loss. The privacy account theorem tells us that for a fixed $r = \sigma^2/n$, privacy loss is smaller for σ and m both being small. An upper bound on the distortion rate r and a lower bound on the number of sub-batched clients would dominate the choice of σ. But, since the data in federated settings is non-IID and the data contributed from various clients might be very distinct. We can therefore define the between clients variance V_c as a measure of similarity between client updates.

We investigate differential privacy in the simulated federated learning setting for different $N \in 100, 1000, 10000$. In each setting the clients get exactly 600 data points.

For all three N, we perform a cross-validation iteration search on the following parameters:

—Number of batches per client B

—Epochs to run on each client E

—Number of clients participating in each round m

—The GM parameter σ

6. EXPERIMENTS

We divide the MNIST raw data into parts. Consequently each client gets two parts, which from the probability theorem, averagely most clients will take test samples from two digits only. Therefore we construct the system that every single client could never train a model properly with only local data to train the model which can predict all ten digits.

6.1 Data preprocessing

We mainly use MNIST dataset to compose our training and validation set, and we sort the query for every label digit. Initially, we construct several simulated clients on the single node. And we split the raw training data to several parts, and every two parts serve a client. Finally, every client would receive about 500 records of training data, and the shuffle process is implemented by random permutation in order to ensure every client holds training data with different digits.

6.2 Training graph construction

First, we construct two operations to increase the global step variable and set global step variable respectively. The loss and correct evaluation is defined in the MNIST graph, where we construct the fully connected model with 2 hidden layers. In each round, if the last step exhausts the privacy budget, then we will enforce to end the round. For each round, we implement the random choosing process by those pre-constructed files storing the possible sub batch each client will use in each communication round in quite a manual way, including selecting and shuffling.

Then, we construct the $model\_placeholder$ to store the new placeholder for the total model, derived from the local model variables. Also, we need the assignment operations to update the model variables to the local ones. $load\_from\_directory\_or\_initialize$ function helps us to initialize our model, where we construct the $accuracy\_account$ and $delta\_account$ or we check whether those model records already exist. The $noiseaccount$ is also constructed in this function.

6.3 Result and analysis

Because of the mechanism adopted by Federated learning, as Fig.2 and Fig.3 shows, Federated learning always sacrifices its performance in the aspect of accuracy. Intuitively, this approach only utilizes part of the training data at each step and non-IID data property can be another important fact for lower performance.

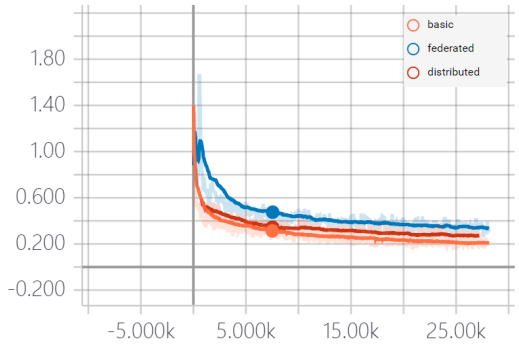

Fig. 2. Training loss under different frameworks

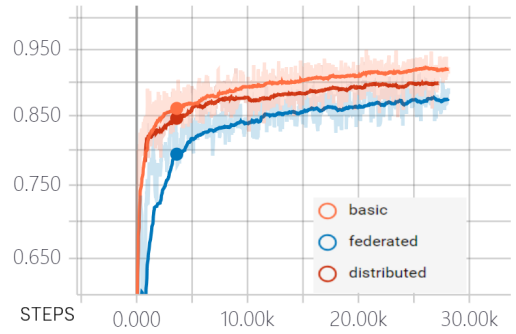

Fig. 3. Validation accuracy under different frameworks

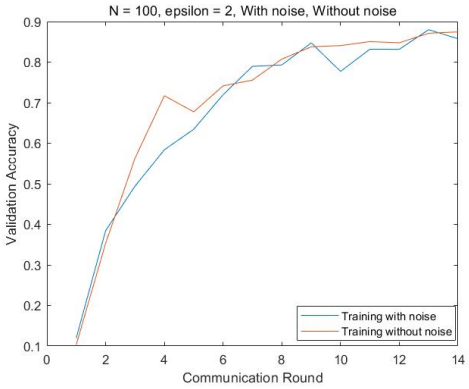

Fig. 4. Validation accuracy under different noise conditions

However, in the Federated settings, as Fig.4 shows, noise mechanisms being differentially private won't effect much of the performance in the initial few steps. For a certain number of clients, if we specify the number of clients participated in each communication round, then in the first few rounds, as Fig.5 shows, the privacy loss tends to increase exponentially, but the model accuracy won't get much improvement during the federated training.

Given the privacy budget, if we set larger ϵ, it can be inferred that the model will be more tolerant to the privacy loss hazard. Actually,

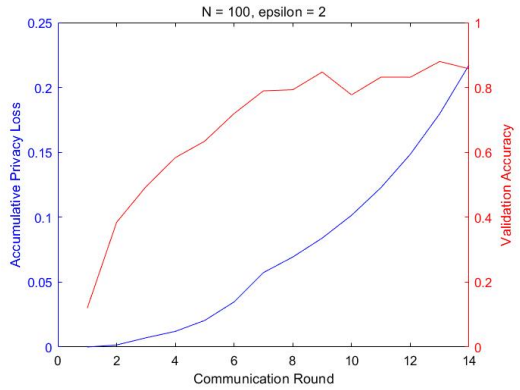

Fig. 5. Relationship between privacy loss and accuracy with 100 clients

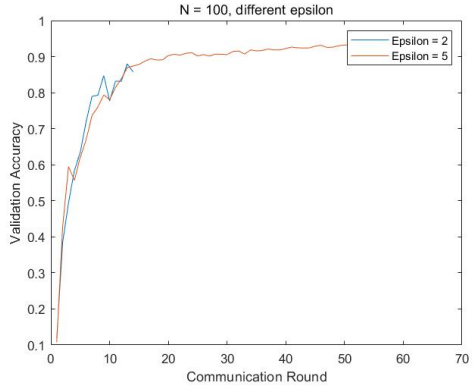

Fig. 6. Validation accuracy with different privacy tolerance

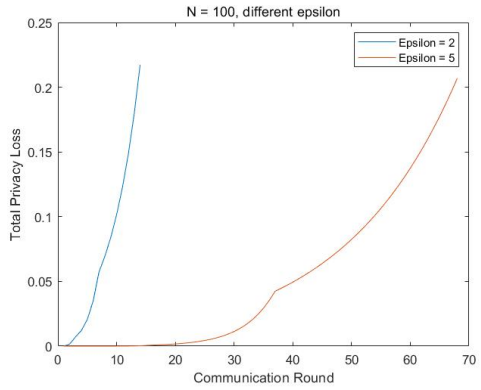

Fig. 7. Privacy loss with different privacy tolerance

if we set the ϵ to be infinite, the differentially private model would totally degrade to the non-private training system.

Clients participated in each communication round would also affect the privacy loss among them, as Fig.8,9 shows.

7. CONCLUSION

To deal with the privacy concern in edge computing, we show the effectiveness of federated learning and compare the results with

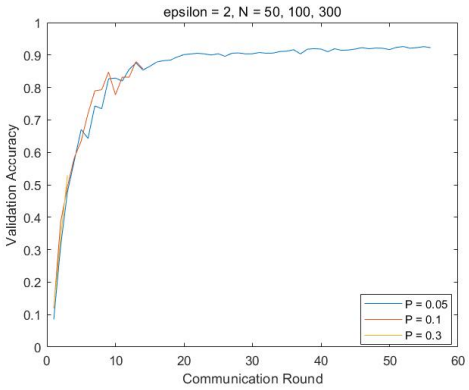

Fig. 8. Larger participation ratio would terminate communication step in advance

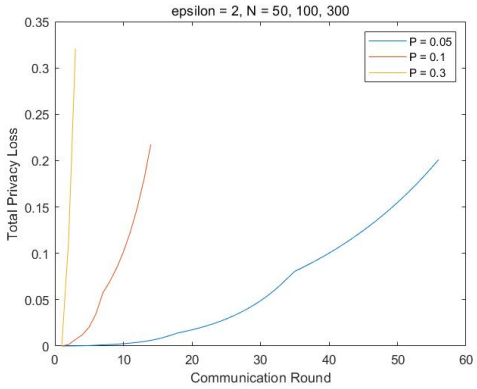

Fig. 9. Larger participation ratio would terminate privacy loss accumulation in advance

the centralized and distributed structure on the image classification task. To improve federated model with respect to privacy protection, we introduce the differential privacy mechanism into model training. By testing the models with different noise approximation levelscomparing them with the original model and analyzing the privacy loss quantitatively in various communication rounds and client number, we show the accuracy could be preserved with more effective privacy protection and there are many works to do to analyze and reduce the privacy loss in the context of edge computing.

8. TYPICAL REFERENCES IN NEW ACM REFERENCE FORMAT

An arXiv article [Hard et al. 2018], an arXiv article [Brendan McMahan Eider Moore Daniel Ramage Seth Hampson Blaise AgüeraAg et al. 2016], an arXiv article [Geyer et al. 2017], a monograph [Goos et al.], a technical report [Dwork], an arXiv article [He et al. 2017], an arXiv article [Abadi et al. 2016], a monograph[Dwork et al. 2014].

REFERENCES

Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. (jul 2016). DOI:http://dx.doi.org/10.1145/2976749.2978318

H Brendan McMahan Eider Moore Daniel Ramage Seth Hampson Blaise AgüeraAg, Agüera Arcas, H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. (feb 2016). https://arxiv.org/pdf/1602.05629.pdf

Cynthia Dwork. *A Firm Foundation for Private Data Analysis*. Technical Report. http://www.cs.bilkent.edu.tr/

C Dwork, A Roth, Cynthia Dwork, and Aaron Roth. 2014. *The Algorithmic Foundations of Differential Privacy*. Vol. 9. 211–407 pages. DOI:http://dx.doi.org/10.1561/0400000042

Robin C. Geyer, Tassilo Klein, Moin Nabi, Sap Se, and Eth Zurich. 2017. Differentially Private Federated Learning: A Client Level Perspective. (dec 2017). https://arxiv.org/pdf/1712.07557.pdf

Gerhard Goos, Juris Hartmanis, Jan Van, Leeuwen Editorial Board, David Hutchison, Takeo Kanade, Josef Kittler, Jon M Kleinberg, Friedemann Mattern, Eth Zurich, John C Mitchell, Moni Naor, Oscar Nierstrasz, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y Vardi, and Gerhard Weikum. *LNCS 3876 - Theory of Cryptography*. https://link.springer.com/content/pdf/10.1007

Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated Learning for Mobile Keyboard Prediction. *arXiv1811.03604v1 [cs]* (2018), 0–6. http://arxiv.org/abs/1811.03604v1

Jianping He, Lin Cai, and I T Mar. 2017. Differential Private Noise Adding Mechanism and Its Application on Consensus . (2017), 1–11.