

PRIVATE WIRELESS FEDERATED LEARNING WITH ANONYMOUS OVER-THE-AIR COMPUTATION

Burak Hasircioğlu and Deniz Gündüz

Department of Electrical and Electronic Engineering, Imperial College London, UK

ABSTRACT

In conventional federated learning (FL), differential privacy (DP) guarantees can be obtained by injecting additional noise to local model updates before transmitting to the parameter server (PS). In the wireless FL scenario, we show that the privacy of the system can be boosted by exploiting over-the-air computation (OAC) and anonymizing the transmitting devices. In OAC, devices transmit their model updates simultaneously and in an uncoded fashion, resulting in a much more efficient use of the available spectrum. We further exploit OAC to provide anonymity for the transmitting devices. The proposed approach improves the performance of private wireless FL by reducing the amount of noise that must be injected.

Index Terms— federated learning, differential privacy, anonymity

1. INTRODUCTION

Federated learning (FL) [1] allows multiple devices, each with its own local dataset, to train a model collaboratively with the help of a parameter server (PS) without sharing their datasets. At each iteration of FL, the PS shares the most recent global model with the devices, which then use their local datasets to update the global model. Local updates are aggregated and averaged at the PS to update the global model. The fact that data never leaves the devices is considered to protect its privacy. However, recent works [2–4] show that model updates or gradients also leak a lot of information about the dataset. This calls for additional mechanisms to guarantee privacy. Differential privacy (DP) is a widely-adopted rigorous notion of privacy [5]. Given an algorithm whose output is some statistics about a dataset, if the change in the output probability distribution is tolerably small, when the input database is changed with a very close neighbouring one, then the algorithm is deemed differentially private. Many recent works exploit DP-based algorithms to provide rigorous privacy guarantees in machine learning [6–11].

We consider co-located devices communicating with the PS over a wireless multiple access channel (MAC). Recent work has shown that rather than the conventional digital

communication, the devices can transmit their local updates simultaneously in an uncoded fashion to enable over-the-air computation (OAC) [12–14]. This results in a much more efficient use of the available resources, and significantly improves the learning performance. It is also remarked in [13] that the analog computation in a wireless domain makes privacy more easily attainable. A common method to provide DP guarantees is adding noise to the output with a variance proportional to the maximum change in the output under neighbouring datasets [5]. In the digital implementation of FL, where each device separately communicates with the PS, noise is added to the gradients after each local iteration [9] to ensure DP; whereas in analog computation, the PS only sees the sum of the updates together with the channel noise, which effectively protects all the updates. Thus, the same amount of protection can be achieved with less perturbation, improving also the final accuracy.

Privacy in wireless FL through OAC has been recently studied in several works [15–18]. In [15], distributed stochastic gradient descent (SGD) is studied with quantized gradients and privacy constraints. In [16], if the channel noise is not sufficient to satisfy the DP target, a subset of the devices add additional noise to their updates, benefiting all the devices. In [17] and [18], transmit power is adjusted for the same privacy guarantee. We note that, in [16–18], channel state information (CSI) at the devices is crucial not only to align their computations at the PS but also for the privacy guarantee. However, in practice, devices acquire CSI from pilots transmitted by the PS. Hence, as an adversary, the PS can adjust the pilots to degrade the privacy level. Additionally, in [16], devices depend on others for privacy, which introduces additional point-of-failure.

Differently from the works cited above, we provide privacy in wireless FL by exploiting the anonymity of the transmitters in OAC. Our main contributions can be summarized as follows: (1) We study the effects of randomly sampling the devices participating in each iteration, and batching in local datasets on the privacy. By forcing a constant receive power at the PS across iterations, we provide anonymity to the transmitting devices, and employ it for privacy; (2) By distributing the noise generation process across the workers, we make the privacy guarantee resilient against the failure of transmitting nodes; (3) Our protocol is robust against pilot attacks.

This work was partially supported by the European Research Council (ERC) through project BEACON (No. 677854).

2. PRELIMINARIES

Before presenting the system model and the proposed solution, we give some preliminaries on DP and the threat model.

2.1. Differential Privacy (DP)

Definition 1. (ϵ, δ) -DP [5]: A randomized mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ is (ϵ, δ) -differentially private (DP) if

$$\Pr(\mathcal{M}(D) \in \mathcal{S}) \leq e^\epsilon \Pr(\mathcal{M}(D') \in \mathcal{S}) + \delta,$$

$\forall \mathcal{S} \subseteq \mathcal{R}$ and $\forall D, D' \in \mathcal{D}$ such that $\|D - D'\|_1 \leq 1$, which is defined, in this work, as the two very close datasets with the same cardinality differing only in one element. (ϵ, δ) -DP is a characterization of the change in the output probability distribution of \mathcal{M} under small changes in the input dataset. ϵ characterizes the privacy loss under such a small change and δ is the upper bound on the failure probability of the bound $\Pr(\mathcal{M}(D) \in \mathcal{S}) \leq e^\epsilon \Pr(\mathcal{M}(D') \in \mathcal{S})$. Thus, small ϵ and δ are desired.

Definition 2. (α, ϵ) -Rényi DP [19]: Rényi divergence between two probability distributions is defined as

$$D_\alpha(P||Q) \triangleq \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q(x)} \left(\frac{P(x)}{Q(x)} \right)^\alpha.$$

A randomized mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ satisfies (α, ϵ) -Rényi DP (RDP) if $D_\alpha(\Pr(\mathcal{M}(D) = x) || \Pr(\mathcal{M}(D') = x)) \leq \epsilon$, where $\alpha \in [1, \infty)$, $\forall D, D' \in \mathcal{D}$ such that $\|D - D'\|_1 \leq 1$. $D_\alpha(\cdot || \cdot)$ for $\alpha = 1$ and $\alpha = \infty$ are defined by continuity.

Definition 3. Gaussian Mechanism (GM): A mechanism \mathcal{M} , which alters the output of another algorithm $f : \mathcal{D} \rightarrow \mathcal{R}$, by adding Gaussian noise, i.e., $\mathcal{M}(D) \triangleq f(D) + \mathcal{N}(0, \sigma^2)$ is called a GM. GM satisfies (ϵ, δ) -DP [5] with $\epsilon = \sqrt{2 \log(1.25/\delta)} \Delta f / \sigma$ and $\Delta f \triangleq \max_{D, D' \in \mathcal{D}} \|f(D) - f(D')\|_2$. It also satisfies (α, ϵ') -RDP with $\epsilon' = \alpha(\Delta f)^2 / (2\sigma^2)$ [19].

Lemma 1. *RDP to DP [19]: A mechanism satisfying (α, ϵ) -RDP also satisfies $(\epsilon + \log(1/\delta)/(\alpha - 1), \delta)$ -DP.*

Lemma 2. *Composition of GM under RDP [19]: Composition of T RDP mechanisms operating on the same dataset with common α and with $(\epsilon_1, \epsilon_2, \dots, \epsilon_T)$ gives an $(\alpha, \sum_{i=1}^T \epsilon_i)$ -RDP mechanism.*

As seen in Lemma 2, composition under Rényi DP is very simple to compute for GM. Moreover, it gives a much tighter bound on the final ϵ value compared to the advanced composition theorem [5], which is used in [16, 18].

2.2. Threat Model

We assume that the PS is semi-honest and curious. It is honest in the sense that it wants to acquire the final trained model, and thus, follows the model averaging step accurately, but it also would like to acquire all possible sensitive information about single individual data points. Therefore, it can attack

the CSI estimation process to encourage the devices to increase their transmit power or to add less noise to their transmissions by suggesting that their channel quality is worse than it is in reality. We note that, if a common pilot signal is used to learn the CSI by all the devices, the CSI values can only be scaled by the same parameter across the devices.

We assume that the devices are honest and trusted. They do not intentionally attack the learning process; however, we also do not depend on a few devices for privacy as the devices may fail to transmit unintentionally. We further assume that the PS has a single receive antenna, thus it is not capable of determining the identities of the transmitting devices through transmission directions.

3. SYSTEM MODEL

We consider N wireless devices. Each device $i \in [N] \triangleq \{1, 2, \dots, N\}$ hosts their local dataset D_i , which are i.i.d. across the devices. A PS orchestrates these devices to learn a global model \mathbf{w} by minimizing a global loss function $\mathcal{L} = \frac{1}{N} \sum_{i \in [N]} \frac{1}{|D_i|} \sum_{d \in D_i} \ell(\mathbf{w}, d)$, where $\ell(\mathbf{w}, d)$ is an application-specific empirical loss function. We assume that the dataset is distributed among the devices with the identical probability and independent of each other. We employ distributed SGD to iteratively minimize \mathcal{L} across the devices. At the beginning of iteration t , the PS broadcasts the global model \mathbf{w}_t to all the devices. Then, the devices participating in the learning process in that iteration transmit their gradient estimates over a Gaussian MAC. In round t , the PS receives the superposition of the signals transmitted by the participating devices as

$$\mathbf{y}[t] = \sum_{i \in A_t} c_{i,t} \mathbf{x}_i[t] + \mathbf{z}[t],$$

where $\mathbf{z}[t] \sim \mathcal{N}(0, N_0 \mathbf{I})$ is the channel noise, A_t is the set of participating devices, $\mathbf{x}_i[t]$ is the signal transmitted by device i , and $c_{i,t}$ is the channel coefficient from device i to the PS. We assume that the transmitters perfectly know and correct the phase shift in their channels. Therefore, for simplicity, we assume real channel gains in the rest of the paper, i.e., $c_{i,t} \in \mathbb{R}^+$.

We assume that $c_{i,t}$ is independent across the users and rounds, but remains constant within one round. We further assume that only the agents with sufficiently good channel coefficients participate in each round to increase power efficiency. We assume this happens with probability $p \in (0, 1)$ for each device, independently of other devices. We leave the analysis of this probability for different fading distributions as future work. If the device i participates in a round, then it samples uniformly from its own local dataset such that every data point is sampled independently from the other samples with probability $q \in (0, 1)$. Let $B_{i,t}$ denote the set of samples in the batch used by the i th device if it is participating in round t . Let $a_t \triangleq |A_t|$ and $b_{i,t} \triangleq |B_{i,t}|$. Both a_t and $b_{i,t}$ are binomial random variables with parameters p and q .

Each participating device computes a gradient estimate based on the random samples in its batch. We bound L_2 -norm of every *per-sample* gradient by L . If this is not the case originally, then the gradient is downscaled by a factor of $\frac{L}{\|\nabla\ell(\mathbf{w}_t, j)\|_2}$; that is, for a sample d , we define $\mathbf{g}_t(d) \triangleq \nabla\ell(\mathbf{w}_t, d) \times \max\left\{1, \frac{L}{\|\nabla\ell(\mathbf{w}_t, d)\|_2}\right\}$, as done in [9].

Device i transmits

$$\mathbf{x}_i[t] = h_{i,t} \left(\xi_{i,t} \sum_{j \in B_{i,t}} \mathbf{g}_t(j) + \beta_{i,t} \mathbf{n}_{i,t} \right), \quad (1)$$

where $\mathbf{n}_{i,t} \sim \mathcal{N}(0, \sigma_{i,t}^2 \mathbf{I})$, $\xi_{i,t}$ and $\beta_{i,t}$ are scaling factors, and $h_{i,t} \triangleq (\tilde{c}_{i,t})^{-1}$. $\tilde{c}_{i,t}$ is the CSI which can be manipulated by the PS by a multiplicative factor of $k \in (0, 1]$, i.e., $\tilde{c}_{i,t} = k \cdot c_{i,t}$. For simplicity, we assume that the local datasets are of the same size, although our analysis can be extended easily by adding another scaling factor to the gradient, if this is not the case.

The PS updates the model as $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{y}[t]$, where η is the learning rate. We assume that the number of participating devices a_t and the batch sizes $b_{i,t}$ are known to the devices but not to the PS. This can be achieved by keeping a common random number generator state across the devices, or alternatively by encrypted communication between the devices. Since this is only sharing two real numbers, the communication overhead is limited.

4. MAIN RESULTS

As we discussed in the previous section, we sample the participating devices and the local datasets to be used in each iteration. It is well known that the privacy of randomized mechanisms can be amplified by subsampling [20–22]; however, these results are for the centralized setting. To take advantage of sampling in the distributed setting, we make sure that all of the data samples are chosen independently and uniformly with probability pq . However, the challenge is that since the local datasets of different devices are distinct, the conditional probability of a data point being chosen given some other data point is already sampled may not be the same as the marginal probability of the same event. For example, if these two data points are hosted by the same device, then the conditional probability of the second data point being chosen is q , instead of pq . One way of overcoming this problem is shuffling the whole dataset across the devices after each iteration to cancel the effect of locality. However, this would incur a large communication cost. A better way is to exploit the wireless nature of the protocol as we explain next.

4.1. Ensuring the anonymity of devices

Since the PS can only see the aggregated signal, it cannot know which devices have transmitted. Still, if the PS knows the number of participating devices, then it can collect some information across the iterations and infer some dependency between the samples, such as two samples being hosted by

the same device. In such cases, the i.i.d. assumption on the sampling probability of data points does not hold. Next, we show how we mask the number of participating devices.

Lemma 3. *If $\xi_{i,t} = 1/b_t, \forall i \in A_t$, where $b_t \triangleq \sum_{i \in A_t} b_{i,t}$, the PS cannot infer the number of devices actively transmitting in round t .*

Proof. Since the devices scale their local gradients by b_t before transmitting, and the PS can only see the average gradient at each round, the received power level at the PS is independent of the number or identity of the transmitting devices. Thus from the power level, it is not possible to infer any information about the number of transmitting devices. \square

Due to Lemma 3, learning any set of sampled data points does not convey any information to the PS about the remaining sampled data points unless it learns the identities of the transmitting devices. Therefore, batching local samples independently with probability q and similarly sampling the transmitting devices independently with probability p is equivalent to subsampling in the centralized setting with probability pq .

4.2. Robustness to transmission failures

As we have stated, we want our scheme to be robust against transmission failures of devices that are scheduled to transmit at a certain iteration, but failed to do so for some reason. This is achieved by distributing the noise generation across the devices.

Lemma 4. *If we choose $\beta_{i,t} = 1/\sqrt{a_t}$ and $\sigma_{i,t} = \hat{\sigma}_t, \forall i \in A_t$, then the received signal at the PS becomes*

$$\mathbf{y}[t] = \sum_{i \in A_t} \frac{1}{b_t} \sum_{j \in B_{i,t}} \mathbf{g}_t(j) + \mathbf{n}[t] + \mathbf{z}[t], \quad (2)$$

where $\mathbf{n}[t] \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I})$, and $\sigma_t = \hat{\sigma}_t$. When $k < a_t$ devices in A_t fail to transmit, the noise variance degrades to $\sigma_t = \hat{\sigma}_t \sqrt{(a_t - k)/a_t}$.

Proof. If k devices fail, there are $a_t - k$ remaining devices transmitting. Since the noise added by the devices are independent and each has the variance $\hat{\sigma}_t^2/a_t$, we have $\sigma_t^2 = \hat{\sigma}_t^2(a_t - k)/a_t$. If there is no failure, $k = 0$, and we obtain $\sigma_t = \hat{\sigma}_t$. \square

As we see in Lemma 4, in case of transmission failures in A_t , the variance of the total noise degrades gracefully, and so does the privacy.

4.3. Robustness to manipulated CSI values

In practice, given a channel noise variance N_0 , we need to tune σ so that the total noise variance $N_0 + \sigma^2$ is sufficient to meet the desired privacy level (ϵ, δ) . However, if we rely on the channel noise and our channel is better than we think, then the total noise variance may be less than it should be, resulting in larger ϵ and δ values than the desired privacy level. This may cause a privacy breach. To avoid this, in the privacy

computations, we simply ignore the channel noise since CSI values are prone to attacks by the PS. Then, the ϵ value we get is the upper bound of the real ϵ value in the presence of channel noise.

4.4. Privacy analysis

In the next theorem, we present a result which shows the boosting effect of sampling on the overall privacy.

Theorem 1. *Each round t in our FL scheme with OAC is $(\alpha, \frac{2p^2q^2\alpha}{\tilde{\sigma}_t^2})$ -RDP, where $\tilde{\sigma}_t \triangleq \frac{\sigma_t b_t}{2L}$, if $pq \leq 1/5$, $\tilde{\sigma}_t \geq 4$ and the following inequalities are satisfied*

$$1 < \alpha \leq \frac{1}{2} \tilde{\sigma}_t^2 \log \left(1 + \frac{1}{pq(\alpha-1)} \right) - 2 \log(\tilde{\sigma}_t), \quad (3)$$

and

$$\alpha \leq \frac{\frac{1}{2} \tilde{\sigma}_t^2 \log^2 \left(1 + \frac{1}{pq(\alpha-1)} \right) - \log 5 - 2 \log(\tilde{\sigma}_t)}{\log \left(1 + \frac{1}{pq(\alpha-1)} \right) + \log(pq\alpha) + \frac{1}{2\tilde{\sigma}_t^2}}. \quad (4)$$

Proof. Consider (2) and define $\mathbf{f}[t] \triangleq \mathbf{y}[t] - \mathbf{n}[t] - \mathbf{z}[t] = \frac{1}{b_t} \sum_{i \in A_t} \sum_{j \in B_{i,t}} \mathbf{g}_t(j)$. Remember that $\Delta f = \max_{D, D'} \|f(D) - f(D')\|_2$, where D and D' are two datasets with the same size, differing only in one data point. For the same realization of random batching and device selection processes, $B_{i,t}(D)$ and $B_{i,t}(D')$ also differ at most in one element. Since $\|\mathbf{g}_t(j)\|_2 \leq L$, we find $\Delta f = \frac{2L}{b_t}$. With GM, according to Definition 3, our mechanism is $(\alpha, \alpha(\Delta f)^2 / (2\sigma_t^2))$ -RDP without sampling. Moreover, by Lemma 3, the sampling probability of each sample is pq . The remaining of the lemma follows from direct application of [22, Theorem 11]. \square

If the conditions in Theorem 1 are not satisfied, the RDP of the sampled Gaussian mechanism can still be computed numerically and improvement due to sampling is still valid. The numerical procedure for this is given in [22, Section 3.3] and it is demonstrated in Section 5.

Process to compute (ϵ, δ) -DP after T iterations: According to Lemma 2, if the received message at round t is (α, ϵ_t) -RDP, then after T iterations, the mechanism is $(\alpha, \sum_{t \in [T]} \epsilon_t)$ -RDP. According to Lemma 1, it is $(\sum_{t \in [T]} \epsilon_t + \log(1/\delta)/(\alpha-1), \delta)$ -DP. We observe that although ϵ_t depends on the parameters in our mechanism, α does not. It is a parameter we choose to minimize $(\sum_{t \in [T]} \epsilon_t + \log(1/\delta)/(\alpha-1), \delta)$, which we compute for several α values and take the best α among them. Since both the analytical (if it exists) and the numerical computations of the composed (α, ϵ) -RDP is computationally cheap, this is feasible.

5. NUMERICAL RESULTS

In this section, we numerically calculate the composite (ϵ, δ) -DP of a learning task with $\delta = 10^{-5}$, for different sampling rates given by pq , where $pq = 1$ means there is no sampling, i.e., all the devices participate and they use all their

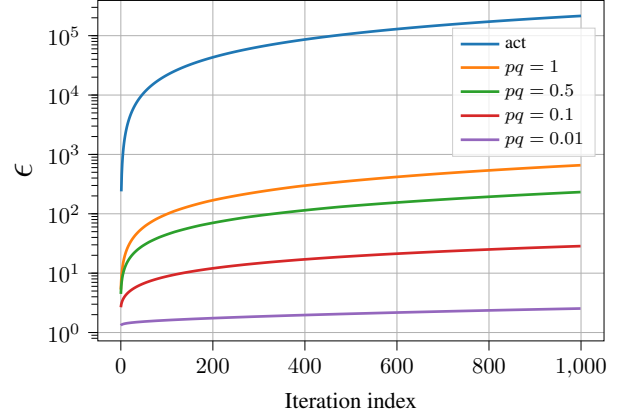


Fig. 1. Total privacy measured by ϵ for different sampling rates across iterations.

local datasets. We assume $\Delta f = 1$ and $\sigma_t = 1, \forall t \in [T]$. For $pq = 1$, we compute the total privacy spent, i.e., the composite ϵ by using both the advanced composition theorem [5, Theorem 3.20], which is denoted by ‘act’ in the figure, and the RDP approach given in Lemma 2. Since we numerically verify that RDP accounting performs much better for the same sampling rate, we use it for the other values of $p \cdot q$. For numerically computing the composition with RDP, we used Opacus library¹. We tested α values from 1 to 64, and picked the one minimizing the composite ϵ value.

We observe in Fig. 1 that for $pq = 1$ and $pq = 0.5$, the resultant ϵ values are very high, namely more than 100. This is, in fact, very weak privacy. Roughly speaking, it means that the output distribution of the final learned model can change by a factor of 100 when only one element is changed in the dataset. However, we see that when we have smaller sampling rates, the privacy guarantee comes into an acceptable range. In an edge network setting, we expect many devices to participate in learning, with substantial local datasets. Therefore, pq values on the order of 0.01 are not unreasonable.

6. CONCLUSION

We have exploited the anonymity in wireless transmissions to enable private wireless FL across edge devices. This is achieved by exploiting OAC rather than orthogonal transmissions, which would reveal the identity of the devices. In particular, we exploit random subsampling of both the devices and the local data samples in achieving reasonable DP guarantees. As opposed to recent works on the topic, we do not depend on the channel gain or the noise at the PS for privacy guarantees as these values are prone to attacks; although our method is orthogonal to these techniques and can be combined with them. Our results demonstrate yet another favourable property of OAC in wireless edge learning. We will evaluate the benefits of the proposed privacy mechanism and its comparison with other alternatives on a practical learning problem in our future work.

¹<https://github.com/pytorch/opacus>

7. REFERENCES

- [1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.
- [2] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov, “Exploiting unintended feature leakage in collaborative learning,” in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.
- [3] Ligeng Zhu, Zhijian Liu, and Song Han, “Deep leakage from gradients,” in *Advances in Neural Information Processing Systems*, 2019, pp. 14774–14784.
- [4] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller, “Inverting gradients—how easy is it to break privacy in federated learning?,” *arXiv preprint arXiv:2003.14053*, 2020.
- [5] Cynthia Dwork, Aaron Roth, et al., “The algorithmic foundations of differential privacy,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.
- [6] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova, “Rappor: Randomized aggregatable privacy-preserving ordinal response,” in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 1054–1067.
- [7] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu, “Collecting and analyzing multidimensional data with local differential privacy,” in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 638–649.
- [8] Shaowei Wang, Liusheng Huang, Yiwen Nie, Xinyuan Zhang, Pengzhan Wang, Hongli Xu, and Wei Yang, “Local differential private data aggregation for discrete distribution estimation,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 2046–2059, 2019.
- [9] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.
- [10] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang, “Learning differentially private recurrent language models,” *arXiv preprint arXiv:1710.06963*, 2017.
- [11] Robin C Geyer, Tassilo Klein, and Moin Nabi, “Differentially private federated learning: A client level perspective,” *arXiv preprint arXiv:1712.07557*, 2017.
- [12] Mohammad Mohammadi Amiri and Deniz Gündüz, “Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.
- [13] Mohammad Mohammadi Amiri and Deniz Gündüz, “Federated learning over wireless fading channels,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.
- [14] Guangxu Zhu, Yong Wang, and Kaibin Huang, “Broadband analog aggregation for low-latency federated edge learning,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.
- [15] Amir Sonee and Stefano Rini, “Efficient federated learning over multiple access channel with differential privacy constraints,” *arXiv preprint arXiv:2005.07776*, 2020.
- [16] M. Seif, R. Tandon, and M. Li, “Wireless federated learning with local differential privacy,” in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2604–2609.
- [17] Yusuke Koda, Koji Yamamoto, Takayuki Nishio, and Masahiro Morikura, “Differentially private aircomp federated learning with power adaptation harnessing receiver noise,” *arXiv preprint arXiv:2004.06337*, 2020.
- [18] Dongzhu Liu and Osvaldo Simeone, “Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control,” *arXiv preprint arXiv:2006.05459*, 2020.
- [19] Ilya Mironov, “Rényi differential privacy,” in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 2017, pp. 263–275.
- [20] Borja Balle, Gilles Barthe, and Marco Gaboardi, “Privacy amplification by subsampling: Tight analyses via couplings and divergences,” in *Advances in Neural Information Processing Systems*, 2018, pp. 6277–6287.
- [21] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan, “Subsampled rényi differential privacy and analytical moments accountant,” in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1226–1235.
- [22] Ilya Mironov, Kunal Talwar, and Li Zhang, “Rényi differential privacy of the sampled gaussian mechanism,” *arXiv preprint arXiv:1908.10530*, 2019.