Report on Privacy Amplification by Decentralization

Arina Agaronyan
Grace Beyoko
Kathleen Rogan

1. Introduction

In 2020, it was estimated that 1.145 trillion megabytes of data was created every day, with individuals each producing around 2.5 quintillion bytes daily; it is forecast that the amount of data generated will only keep growing. Nowadays, data is used for various different purposes, such as analyzing markets, training, and testing algorithms for machine learning and AI. However, there is a cost in the trade-off between utility and privacy. This has led to an increasing concern over data privacy, notably with the classic centralized approach where a curator is trusted to store and analyses the data.

For the past few years, there has been an increased interest in federate learning combined with the use of differential privacy. In federate learning, the data remains in the hands of their owners, and the results are shared locally with peer-to-peer exchanges, which is an efficient and scalable method. However, local computations can be prone to data leakage or attacks. This is where differential privacy can be useful to perturb the data and ensure privacy.

Differential Privacy was proposed by Dwork in 2006, where data from i participants is considered differentially private if the attacker that has $n-1$ is not able to reliably tell who is n-th participant. An important question is how to enforce privacy without having a significant drop in utility which would make the data completely useless in the case of decentralized paradigms (Dwork, 2006).

A traditional solution that has been used to ensure privacy is local differential privacy (LDP). Each participant adds noise to their data before it is sent to a central coordinator. This offers privacy guarantees, but often degrades the utility of the data due to excessive noise.

To remedy that issue, this paper introduces a new concept called network differential privacy, which consists of a relaxation of LDP – there is no coordinator. It uses the fact that decentralized networks restrict visibility as the data being accessed is limited by the interactions. This allows for an improvement in the privacy-utility trade-offs with a privacy amplification.

2. State of the art

In federate learning, different ways of increasing privacy have been considered; we will see the main ones that are used and how their potential limitations motivated the authors to introduce a novel method. The noise can be added at different levels, either at the server level or at the client level.

2.1. Central Differential Privacy (CDP)

In central differential privacy the noise is added to the parameters that are globally aggregated. This is also called user-level DP – it protects a user's contribution in its entirety. The output is made indistinguishable with a probability that depends on if the user was in the training data or not, and is bounded by ε (Naseri et al., 2022). There are different ways to implement it, but one of the most common ways is to clip the l_2 norm updates and add perturbation to the aggregated model updates (McMahan et al., 2018).

However, this method has its limits as the users have to trust that the server will add noise and update the model, and it is a single point of failure. This method does not necessarily enable a consistent correct trade-off between privacy and accuracy, it mainly depends on the nature of the dataset. Another issue is that privacy loss cumulates as it is accumulated between models updates. It is also a less efficient method when faced with large datasets (Bernau et al., 2021).

2.2. Local differential privacy (LDP)

In general, the most widely used model is Local Differential Privacy. With this method, each user performs a perturbation locally on their own data, which alleviates the need for an intermediary agent. The main advantage is that the users don't need to trust anyone but themselves with their data (Bernau et al., 2021). There are two main mechanisms for LDP, the first one is randomized response, which is the one used in the paper, and the second is unary encoding.

In Randomized response, the user reports the true answer with probability p or a random answer with probability $1-p$. This method is used on binary *yes* or *no* questions. This mechanism was mainly introduced to encourage more honest answers in surveys (Kasiviswanathan et al., 2010).

In Unary encoding, the responses have a defined domain, where the responses are encoded, then a perturbation is

applied, and finally aggregated. This is one of the simpler methods, used for example to build a histogram in the local model (Near & Abuah, 2021).

LDP is a particularly well fit for federal learning. The main drawback of this method is that the accuracy achieved is of lower magnitude than CDP. This is especially the case with very small datasets or those of very high-dimension (Duchi et al., 2013).

2.3. Cryptography

In encryption, there are two main ways to encrypt data for privacy, first of which is Homomorphic encryption, where the data is encrypted using the public key of the aggregator. This supposes that the users trust the aggregator with their data. This can be a issue if the central agent is an attacker (Castelluccia et al., 2009).

The other way is Secure Multi-party Computation (SMC). It is a cryptographic technique where each of the users compute their functions and only share the result. One limit for this method is that it does not stop information from being inferred through the computations. It also requires the users to interact with each other and thus gives rise to risk of collusion (Shi et al., 2011).

In the context of federate learning, cryptographic techniques for privacy are seldom used, as they require an increased computation energy and are not necessarily efficient due to their required interactions either between users or with a central agent.

2.4. DP amplification

They are three main ways amplification of privacy is done. The first method is Amplification by *shuffling*; it is not a privacy model, but more of a layer that can be added. In a local approach, the data is encoded, then grouped and shuffled, which removes the identifying features, before the data is decrypted. However, there is still the issue of collusion if users communicate (Úlfar Erlingsson et al., 2020).

A second method is Amplification by *iteration*. It uses the fact that the more an algorithm is performed repetitively, the less is the privacy risk is for a user, as the individual contribution is diluted. These iterations are performed in a way that information on a single point cannot be retrieved. However, this requires a sufficiently high number of iterations in order to work, which can be costly in terms of computing power, and privacy gains can be limited if the convergence is slow (Feldman et al., 2018).

Finally, Amplification by *subsampling* requires an application of private mechanisms to small random subsamples; it is the most used method out of the three in amplification. It achieves an increase in privacy with less noise, as users have a probability p of being included. This can prove efficient in large datasets. However, the sampling rate has to be selected carefully and users must be independent from each other (Balle et al., 2018).

These techniques are difficult to efficiently put in place in a decentralized setting for different reasons, and are thus extensively used.

In a decentralized context all of these methods have limitations. When the original paper was written, the most employed method was LDP, which had lower score of accuracy than CDP in general, and higher error. Thus, there is a need for a privacy mechanism that would be well suited to decentralization, but would achieve higher results in terms of accuracy capable of rivaling with CDP.

3. Main Results

The paper introduces a decentralized model where users maintain private datasets, and their interactions are limited to neighboring users in a graph. The central insight is that network DP captures privacy from neighboring users and takes into account potential collusion between users, which is not addressed by the classic LDP mechanism.

The work is the first to demonstrate that formal privacy gains can be obtained from a fully decentralized structure, without relying on a central aggregator, while maintaining an efficient privacy-utility trade-off.

3.1. Network DP and Decentralized Model Setup

Let $V = \{1, \dots, n\}$ be a set of n honest but curious users. Each user u has a private dataset D_u, with $D = \{D_1 \bigcup, \dots, \bigcup D_n\}$ as the union of all user datasets. D' represents a dataset that only differs from D by a single user's data.

The model thus assumes a neighboring relation over datasets, where a user's interaction is limited to its neighbors in a graph, thus ensuring that the privacy of each user's dataset is only affected by neighboring users.

The input to the system is D, and the output consists of messages exchanged between users. A message $A(D)$ is of the form (u, m, v), where u sends message m to v. Each user only has access to a restricted view, denoted by $O_u(A(D)) = \{(v, m, v') \in A(D) : v = u \text{ or } v' = u\}$, to maintain and amplify the privacy guarantees of A.

3.2. Token Walk on a Ring

A token moves sequentially along the edges of a directed ring graph with K users. At each node, the token is updated. This model is a first attempt at decentralizing privacy-preserving computation, where the standard deviation σ_{loc}

Algorithm 1 Private real summation on the ring.

$\tau \leftarrow 0$; $a \leftarrow 0$
for $k = 1$ to K **do**
  for $u = 1$ to n **do**
    if $a = 0$ **then**
      $\tau \leftarrow \tau + \text{Perturb}(x_u^k; \sigma_{loc})$
      $a = n - 2$
    else
      $\tau \leftarrow \tau + x_u^k$; $a \leftarrow a - 1$
return τ

Algorithm 2 Private histogram on the ring.

Init. $\tau \in \mathbb{N}^L$ with γn random elements
for $k = 1$ to K **do**
  for $u = 1$ to n **do**
    $y_u^k \leftarrow RR_\gamma(x_u^k)$
    $\tau[y_u^k] \leftarrow \tau[y_u^k] + 1$
for $i = 0$ to $L - 1$ **do**
  $\tau[i] \leftarrow \frac{\tau[i] - \gamma/L}{1-\gamma}$
return τ

Algorithm 3 Private summation on a complete graph.

$\tau \leftarrow 0$, $k_1 \leftarrow 0, \ldots, k_n \leftarrow 0$
for $t = 1$ to T **do**
  Draw $u \sim \mathcal{U}(1, \ldots, n)$
  $k_u \leftarrow k_u + 1$
  $\tau \leftarrow \tau + \text{Perturb}(x_u^{k_u}; \sigma_{loc})$
return τ

Algorithm 4 Private SGD on a complete graph.

Initialize $\tau \in \mathcal{W}$
for $t = 1$ to T **do**
  Draw $u \sim \mathcal{U}(1, \ldots, n)$
  $Z = [Z_1, \ldots, Z_d],\ Z_i \sim \mathcal{N}\big(0, \frac{8L^2 \ln(1.25/\delta)}{\varepsilon^2}\big)$
  $\tau \leftarrow \Pi_{\mathcal{W}}(\tau - \eta(\nabla_\tau f(\tau; D_u) + Z))$
return τ

is added only once every $n - 1$ hops of the token.

Here, Algorithm 1 achieves the same privacy-utility trade-off as a trusted central aggregator. In each k round, the raw contributions of users are aggregated and perturbed before being sent to users, ensuring differential privacy.

Algorithm 2 is applied to discrete histogram computation, where each user's contribution is randomized using randomized response (RR) before being added to the token. The token is initialized with random elements to obscure the first inputs. This is done to leverage results on privacy amplification by shuffling.

Like Algorithm 1, Algorithm 2 offers a privacy gain of $O(1/\sqrt{n})$ compared to LDP, which is more efficient as the number of users increases. Empirical results show that the practical privacy gains are stronger than what is suggested by the theoretical bounds.

While this algorithm outputs an unbiased estimate, satisfying network DP for any $\delta' > 0$, this approach is not robust to collusions. Specifically, if two users collude and share their observations, the algorithm does not satisfy differential privacy. Furthermore, in a fixed ring, one user's privacy would depend on the relative position with another respective user, implying no privacy amplification when one user come directly after the other. This motivates the need for a more flexible graph topology, leading to the exploration of a complete graph.

3.3. Token Walk on a Complete Graph

In this improved model, a complete undirected graph is used, where every pair of distinct nodes is connected. The token is sent to a user chosen uniformly at random, and its path is entirely hidden from all users, ensuring more robust privacy guarantees which rely on intermediate aggregations of values between two visits of the token to a given user, as well as the secrecy of its path.

We analyze the information leakage from the visits of the token. By considering the fictive walk in Algorithm 3 with cycles of size n, we use amplification by subsampling to quantify the privacy loss and show that each cycle incurs a privacy loss of $3\varepsilon/\sqrt{n}$.

The results provide a network DP guarantee of $O(1/\sqrt{n})$ compared to LDP, with improved privacy-utility trade-offs. A more complete model offers improvements upon the LDP as soon as $n \geq 20$. Theoretical results show that this approach asymptotically matches the privacy-utility trade-off of a trusted central aggregator.

On another note, optimizing with Stochastic Gradient Descent (SGD), Algorithm 4 offers privacy amplification of $O(\ln n/\sqrt{n})$ compared to LDP, and the privacy-utility trade-off in nearly the same as that of private SGD in the model with a trusted curator. Empirical results show that the practical privacy gains are stronger than what is suggested by the theoretical bounds.

3.4. Collusion and Robustness

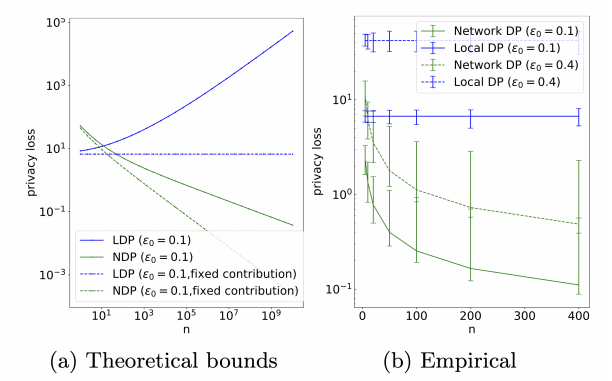
One of the key advantages of the complete graph model is its resistance to collusion. In a complete graph, colluding users can be treated as a single node, thus maintaining privacy even in the presence of malicious users.

The ring graph topology has limitations for gradient descent applications, as the privacy guarantee depends on relative user positions in the ring. In contrast, the complete graph allows for privacy amplification by iteration in gradient descent settings, as privacy grows with the number of gradient steps, irrespective of user positions.

3.5. Experimental Results

Simulations were performed on a random walk of size $T = 100n$, and the privacy loss was computed for all user pairs. The full specifications of the experiment will be outlined later, within our own replication. Results show significant privacy amplification in practice compared to theoretical expectations, especially as the number of users increases.

As previously stated, theoretical analysis shows that network DP outperforms LDP when $n \geq 20$, with privacy loss decreasing as n grows. Empirical results further demonstrate that the privacy gains are stronger in practice than predicted.



(a) Theoretical bounds (b) Empirical

Figure 1. Comparing network and local DP on real summation for $T = 100n$.

The experiments on discrete histogram computation show that network DP provides significant privacy gains, particularly when the number of contributions per user is balanced with the total number of users.

The results demonstrate that network differential privacy (network DP) significantly improves upon local differential privacy in decentralized systems, particularly when collusion is taken into account and the network structure is robust. The proposed algorithms provide strong theoretical privacy guarantees and yield better privacy-utility trade-offs compared to central aggregators, with practical gains observed in simulations. The work represents a significant advancement in the field of decentralized privacy-preserving algorithms and lays the foundation for future research in robust decentralized systems.

4. Critical Analysis

The study offers a unique way to achieve formal privacy gains in fully decentralized systems by leveraging the structure of the network and the local view of the participants. Although the approach is innovative and demonstrates significant results, it is not without limitations. The following section will critically examine the vulnerabilities and merits of the proposed methodology.

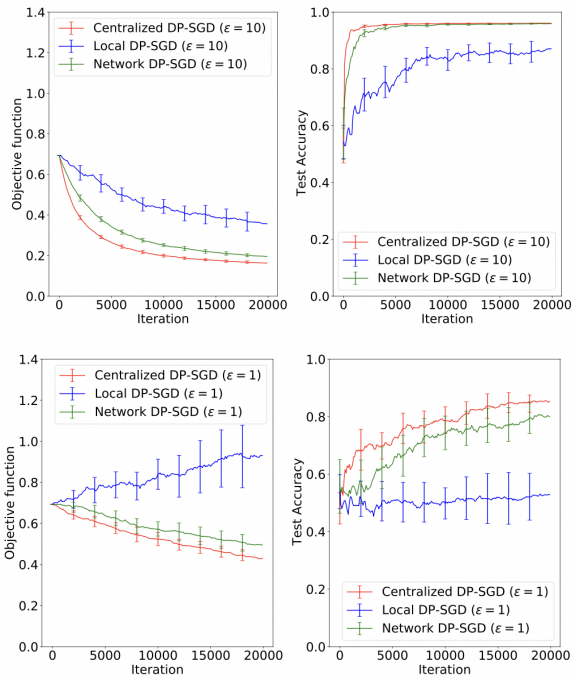


Figure 2. Comparing three settings for SGD with gradient perturbation with different ε values.

4.1. Limitations

Vulnerability to Attacks and Collusion

One of the most significant limitations of the proposed approach is its vulnerability to attacks, particularly label-flipping and data poisoning attacks. As highlighted in "Differential Privacy for Deep and Federated Learning: A Survey" by El Ouadrhiri and Abdelhadi, the fully decentralized protocol proposed by Cyffers et al. relies on a token that sequentially traverses the network, with each node adding noise to its contribution to ensure differential privacy. However, this process is susceptible to malicious actors infiltrating the network. An attacker could manipulate the token by injecting false data or flipping labels, thereby compromising the integrity of the learning process. The authors only state that the algorithm "can tolerate a constant number of collisions at the cost of some reduction in the privacy amplification effect." However, this robustness is conditional and does not fully address the risk of more sophisticated attacks. For instance, if an attacker gains control of multiple nodes or strategically positions themselves within the network, they could significantly degrade the privacy and utility of the system. Furthermore, the assumption that users are "honest-but-curious" (i.e., they follow the protocol but may attempt to infer private information) is a significant limitation. In real-world scenarios, malicious users may not adhere to this assumption, actively seeking to disrupt the learning process or extract sensitive information.

ASSUMPTION OF UNIFORM PARTICIPATION IN RANDOM WALKS

The proposed model relies on the assumption that every node in the network participates equally in the random walk, ensuring that each node has a similar level of interaction and privacy. However, this assumption may not hold in real-world applications. For example, in scale-free networks, such as social networks, certain nodes interact much more frequently than others. In such a scenario, an attacker could focus on highly connected nodes, gaining access to a disproportionate amount of information and compromising the privacy of the entire network. This uneven participation undermines the privacy guarantees of Network DP, as the random walk may not uniformly distribute information across the network. The authors do not address this issue, leaving a gap in the applicability of their approach to real-world decentralized systems where node connectivity and participation are highly variable.

DEPENDENCE ON NETWORK TOPOLOGY

The performance of Network DP is heavily dependent on the underlying network topology. The paper primarily evaluates the approach on two topologies: the ring and the complete graph. While the authors note that the ring topology is "not very robust to collisions," they do not extensively explore the implications of other topologies, such as mesh or tree networks. In practice, the choice of network topology can significantly impact both privacy and utility. For example, in a mesh network, where nodes have multiple connections, the privacy amplification effect may differ from that in a ring or complete graph. Furthermore, the paper does not address the case of dynamic networks, despite acknowledging that "time-evolving topologies can help improve robustness to collusions, in particular in rings and other sparse topologies.". This raises concerns about how well Network DP generalizes to more diverse and dynamic decentralized environments.

AMPLIFICATION LIMITED TO $O(1/\sqrt{n})$

The privacy amplification achieved by Network DP is limited to $O(1/\sqrt{n})$, meaning that the marginal gains in privacy decrease as more participants are added. While larger networks theoretically allow for better privacy amplification, the returns diminish rapidly as n grows. This limits the system's ability to scale effectively in large networks, as adding more nodes provides only marginal improvements in privacy. This amplification limit implies a trade-off between utility and privacy. As n grows, maintaining privacy guarantees may require excessive noise, which reduces the utility of the aggregated data. This trade-off is a significant limitation, particularly in large-scale applications where both privacy and accuracy are critical.

4.2. Merits

PERFORMANCE SIMILAR TO CENTRALIZED MODELS

Despite its limitations, the proposed approach achieves a significant breakthrough by demonstrating that full decentralization can lead to formal privacy enhancements. The authors provide theoretical evidence that Network DP can match the privacy-utility trade-off of centralized models, eliminating the need for a trusted curator while retaining strong privacy guarantees. For instance, in their experiments on stochastic gradient descent (SGD), the authors show that "Network DP-SGD nearly matches the privacy-utility trade-off of Centralized DP-SGD for both $\varepsilon = 1$ and $\varepsilon = 10$ without relying on a trusted curator." This result is particularly noteworthy because it bridges the gap between centralized and decentralized approaches, offering a viable alternative for privacy-preserving machine learning in distributed settings.

IMPROVED PERFORMANCE OVER LDP

One of the most significant contributions of the paper is its demonstration that Network DP significantly outperforms traditional LDP methods. The authors prove that Network DP reduces the error of LDP by a factor of $\sqrt{n}$ for real summation tasks. This improvement is a major advancement in the field, as it addresses one of the key limitations of LDP: its high noise requirements, which often degrade utility in large-scale applications.

ELIMINATION OF SINGLE POINTS OF FAILURE

By eliminating the need for a central server, the proposed approach removes a single point of failure, enhancing the robustness and scalability of privacy-preserving federated learning. This is particularly important in cross-device applications, where the number of participants can be very large, and a central coordinator could become a bottleneck. The authors emphasize that their work is "the first to show that formal privacy gains can be naturally obtained from full decentralization (i.e., from having no central coordinator)." This insight opens new avenues for research and application in decentralized systems, where privacy and scalability are critical concerns.

RESILIENCE TO COLLUSION TO SOME EXTENT

The authors highlight that their approach is "naturally robust to the presence of a (constant) number of colluding users," particularly in the context of a complete graph. This resilience is a notable advantage, as it allows the system to tolerate a limited number of malicious actors without completely compromising privacy. However, this robustness is conditional and depends on the network topology, as the ring topology is less resilient to collusion.

5. Experiments

In this section, we detail our replication of the experiments done in the original paper. The experiments aimed to evaluate the performance of the proposed Network DP framework in comparison to LDP and centralized differentially private models. Specifically, the goal was to train a logistic regression model in a decentralized setting using stochastic gradient descent (SGD) while maintaining strong privacy guarantees.

The experiment compares three variants of private SGD: Centralized DP-SGD, Local DP-SGD, and Network DP-SGD. Centralized DP-SGD refers to the centralized version of differentially private SGD introduced by Shi et al. (2011)(DPS), which assumes the presence of a trusted curator or aggregator. This method serves as a baseline for comparison, as it represents the best-case scenario for privacy-utility trade-offs in a centralized setting. Local DP-SGD, on the other hand, corresponds to Algorithm 4 in the paper, where noise is calibrated to ensure Local Differential Privacy (LDP).

In this setting, each user adds noise to their gradients locally before sharing them, providing strong privacy guarantees but often at the cost of reduced utility. Finally, Network DP-SGD is the proposed decentralized SGD algorithm under Network DP, where noise is calibrated according to the Network DP framework. This approach leverages the decentralized nature of the network to achieve better privacy-utility trade-offs without relying on a trusted central server.

The experiments were conducted using a binarized version of the UCI Housing dataset, obtained from OpenML. The dataset contains a numeric target feature that was converted into a two-class nominal target feature. The conversion was done by computing the mean of the target values: each time the target value was lower than the mean, it was classified as positive, and when above or equal to the mean, it was classified as negative. The dataset was standardized and normalized to ensure that the logistic loss function was 1-Lipschitz, which is a key requirement for the convergence of SGD. The dataset was split into a training set (80%) and a test set (20%), with the training set further divided across n=2000 users, resulting in each user having a local dataset of size 8.

The parameters used in the experiment were carefully chosen to reflect realistic decentralized settings. The number of users (n_nodes) was set to 2000, simulating a large-scale decentralized system. The total privacy budget (ε-tot) was set to 10, representing a low privacy setting, while the privacy relaxation parameter (δ) was set to $1e-6$. The number of iterations was set to 20,000, ensuring that each user contributed multiple times to the model updates. The confidence factor for update limits (conf) and the Lipschitz constant (L) were set to 1.25 and 0.4, respectively. The noise level (σ) was a critical parameter, as it controls the amount of noise added to the model's parameters to protect the privacy of individual data points. For Local DP-SGD, the noise level (σ_{loc}) was computed to be 5.87, while for Centralized DP-SGD, it was significantly lower at 0.70. For Network DP-SGD, the noise level started at 1.97 and gradually decreased over time, reaching a final value of around 1.22. This reduction in noise over time is a key feature of Network DP, as it allows for better privacy-utility trade-offs compared to LDP.

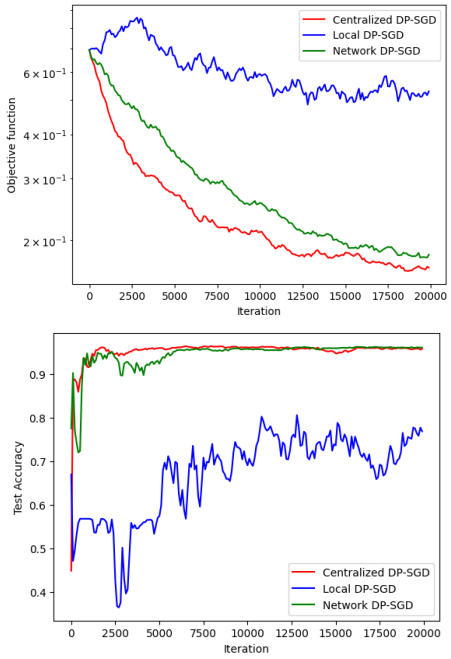


Figure 3. Performances of privacy mechanisms on original dataset

The performance of the algorithms was evaluated using three key metrics: the accuracy of the logistic regression model on the test set, the privacy loss quantified using the (ε, δ)-differential privacy framework, and the standard deviation of the noise added to the gradients. The results demonstrated that Network DP-SGD nearly matches the privacy-utility trade-off of Centralized DP-SGD for $\varepsilon = 10$, without relying on a trusted curator. This is a significant finding, as it shows that formal privacy gains can be achieved in a fully decentralized setting. Additionally, the Network DP-SGD performed slightly better than Local DP-SGD, demonstrating the improved utility of the proposed approach.

The graphs generated from the experiments illustrate these findings. The objective function plot shows that Network DP-SGD converges more effectively than Local DP-SGD, although it does not quite reach the performance of Centralized DP-SGD. This is expected, as Centralized DP-SGD benefits from the presence of a trusted aggregator. The

test accuracy plot further confirms that Network DP-SGD achieves better accuracy than Local DP-SGD, with all three methods showing decreasing accuracy variance over iterations. These results highlight the potential of Network DP as a viable alternative to traditional LDP and centralized differentially private models in large-scale decentralized systems.

One of the main challenges faced during the replication was working with outdated code. The original implementation of the DP-SGD algorithms used a number of deprecated libraries, and the code required extensive modifications to function with modern Python versions and frameworks. Despite these challenges, the replication successfully confirmed the theoretical findings of the paper, demonstrating that Network-DP can achieve privacy-utility trade-offs comparable to centralized models while operating in a fully decentralized setting.

6. Contribution

For our contribution to the results, we decided first to test other datasets to observe the robustness of the results of the new method. And secondly, we wanted to see how modifying different parameters that are important for network DP would impact not only its results, but also the results of the other methods and how they compared.

6.1. New datasets

We tested our code on two other datasets to see how the results would be impacted. The task was still a binary classification task. Firstly, we used a smaller dataset of 4601 instances, the spam dataset of OpenML (Hopkins et al., 1999).

In the upper Figure 4, we can observe that the performances of the network DP-SGD are negatively impacted by a decrease in the size of the dataset, as its performances are worse than the centralized DP-SGD but we still obtain better results than Local DP-SGD

Secondly, we used the Cardiovascular dataset of OpenML (Feurer, 2023) with 70000 instances.

In the lower Figure 4, we can see that with higher-dimension data the Local DP-SGD performances worsen. The network DP-SGP performances are quite close to the ones of centralized and get better with more iterations. A higher-dimension does not seem to impact the network DP which is a sign that it is a scalable method.

6.2. Modification of parameters

We then decided to modify the main parameters to see how it would impact the results on the original dataset.

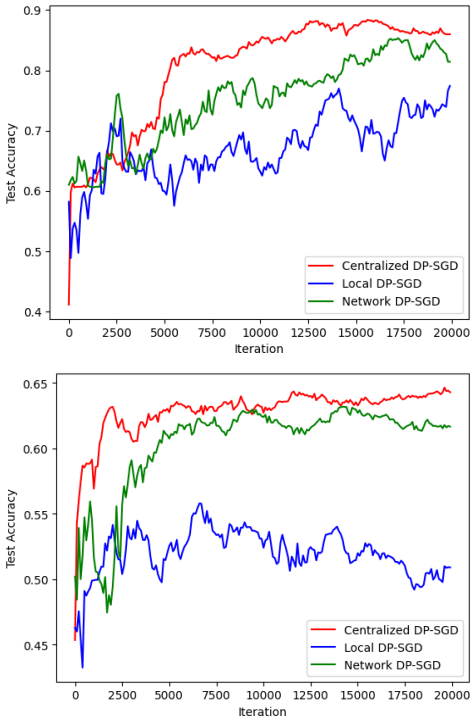


Figure 4. Performances of privacy mechanisms on different datasets

Firstly, we modified the number of updates from 1 to 50 for each node.

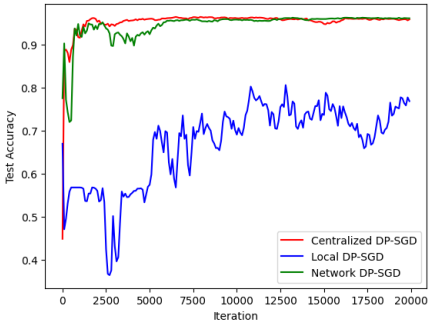


Figure 5. Accuracy with increase of nodes updates

We can see that increasing the updates per nodes reduces very minimally the accuracy and does not have a significant impact.

Secondly, we increased the number of iterations from 2000 to 100000. As visualized in Figure 6, the performances of the network DP improve with an increase in iterations, but the performances of the local DP are negatively impacted by it due to an accumulation of loss, and the performances of central DP stay stable.

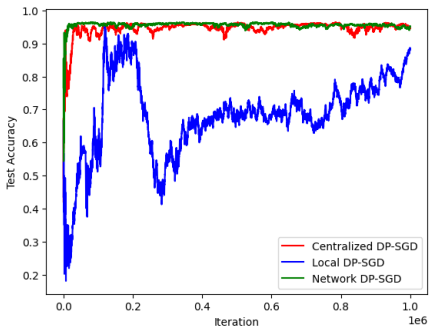


Figure 6. Accuracy with increase of iterations

And finally we changed the number of nodes from 2000 to 200.

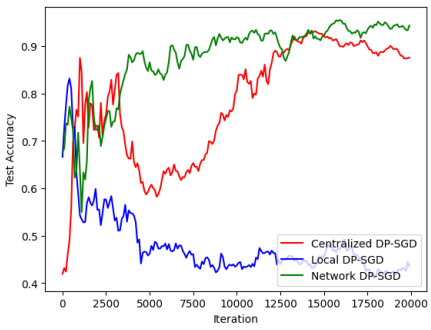


Figure 7. Accuracy with decrease of nodes

Surprisingly, the network DP is impacted only in small proportions compared to central DP, where we can see an important drop in its accuracy. Local DP suffers from an important drop in accuracy from this modification.

7. Conclusion

This paper introduces a novel approach to privacy preservation in decentralized systems through Network DP, which enhances the privacy-utility trade-off by leveraging a decentralized network structure. The method significantly improves upon traditional LDP, which often suffers from high privacy loss and degraded utility due to excessive noise. By utilizing decentralized interactions and network-based privacy amplification, the proposed approach provides stronger privacy guarantees, particularly when considering potential collusion among participants.

The experiments show that, even in the presence of collusion, the complete graph topology offers robust privacy protection while maintaining efficiency and accuracy, especially in applications like federated learning. The results demonstrate that as the number of users grows, Network DP provides better privacy-utility trade-offs compared to LDP, and even approximates the performance of central aggregators, offering a scalable and practical solution.

However, while the findings are promising, several open questions and challenges remain. The reliance on decentralized networks introduces complexity in ensuring privacy when considering malicious behavior or adversarial attacks among users. While the approach demonstrates robustness in the complete graph topology, the privacy guarantees might vary depending on the network structure and the specific interactions between users. Additionally, the efficiency of the method in large-scale systems, especially regarding computational overhead and communication costs, warrants further investigation.

There are also open questions around the scalability of this model in more dynamic environments where user mobility or changes in the network topology could affect privacy outcomes. The potential for privacy leakage in different decentralized settings, particularly with heterogeneous user behavior or data types, also remains an area for further exploration. Moreover, while the proposed method performs well in the experiments conducted, understanding its real-world applicability - especially in highly sensitive domains - requires further empirical validation and testing.

Overall, while Network DP represents a significant advancement in decentralized privacy-preserving algorithms, it opens up multiple avenues for future research to address these questions. By further refining the model to handle diverse and dynamic environments, we can move closer to achieving robust, scalable, and efficient privacy solutions for decentralized systems.

References

Balle, B., Barthe, G., and Gaboardi, M. Privacy amplification by subsampling: Tight analyses via couplings and divergences, 2018. URL https://arxiv.org/abs/1807.01647.

Bernau, D., Robl, J., Grassal, P., Schneider, S., and Kerschbaum, F. Comparing local and central differential privacy using membership inference attacks, 07 2021.

Castelluccia, C., Chan, A. C.-F., Mykletun, E., and Tsudik, G. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(3), June 2009. ISSN 1550-4859. doi: 10.1145/1525856.1525858. URL https://doi.org/10.1145/1525856.1525858.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Local privacy and statistical minimax rates. In *2013 IEEE 54th*

Annual Symposium on Foundations of Computer Science, pp. 429–438, 2013. doi: 10.1109/FOCS.2013.53.

Dwork, C. Differential privacy. In Bugliesi, M., Preneel, B., Sassone, V., and Wegener, I. (eds.), *Automata, Languages and Programming*, pp. 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-35908-1.

Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 521–532. IEEE, October 2018. doi: 10.1109/focs.2018.00056. URL http://dx.doi.org/10.1109/FOCS.2018.00056.

Feurer, M. Cardiovascular-disease-dataset, 2023. URL https://www.openml.org/search?type=data&sort=qualities.NumberOfInstances&status=active&id=45547.

Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. Spambase. UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C53G6X.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately?, 2010. URL https://arxiv.org/abs/0803.0924.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models, 2018. URL https://arxiv.org/abs/1710.06963.

Naseri, M., Hayes, J., and Cristofaro, E. D. Local and central differential privacy for robustness and privacy in federated learning, 2022. URL https://arxiv.org/abs/2009.03561.

Near, J. P. and Abuah, C. Programming differential privacy, 2021. URL https://programming-dp.com/.

Shi, E., Chan, T.-H., Rieffel, E., Chow, R., and Song, D. Privacy-preserving aggregation of time-series data, 01 2011.

Úlfar Erlingsson, Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. Amplification by shuffling: From local to central differential privacy via anonymity, 2020. URL https://arxiv.org/abs/1811.12469.

A. Contribution of each group member

Arina Agaronyan: Main Results, Conclusion, General Editing and Formatting

Grace Beyoko: Critical Analysis, Experiments

Kathleen Rogan: Introduction, State of the Art, Code, Contribution

Link to the jupyter notebook: https://colab.research.google.com/drive/1F-mZceE0lqIe-wg8_KGQDoxm0WMuCaMc?usp=sharing

Link to github: https://github.com/Tiny-boot/privacy-amplification-by-decentralization