Efficient Node Selection in Private Personalized Decentralized Learning

Edvin Listo Zec*[1,2], Johan Östman[3], Olof Mogren[1], and Daniel Gillblad[3]

[1]RISE Research Institutes of Sweden
[2]KTH Royal Institute of Technology
[3]AI Sweden
edvin.listo.zec@ri.se

Abstract

Personalized decentralized learning is a promising paradigm for distributed learning, enabling each node to train a local model on its own data and collaborate with other nodes to improve without sharing any data. However, this approach poses significant privacy risks, as nodes may inadvertently disclose sensitive information about their data or preferences through their collaboration choices. In this paper, we propose Private Personalized Decentralized Learning (**PPDL**), a novel approach that combines secure aggregation and correlated adversarial multi-armed bandit optimization to protect node privacy while facilitating efficient node selection. By leveraging dependencies between different arms, represented by potential collaborators, we demonstrate that PPDL can effectively identify suitable collaborators solely based on aggregated models. Additionally, we show that PPDL surpasses previous non-private methods in model performance on standard benchmarks under label and covariate shift scenarios.

1 Introduction

Collaborative machine learning is a recent paradigm where multiple actors train a joint model without revealing their local datasets [1]. Instead, only the locally trained model parameters are shared among the actors. In applications pertaining to sensitive data, e.g., healthcare and banking, where it may be challenging to collect the data in a single location, collaborative learning has the potential to unlock a plethora of novel collaborations. Collaborative learning is typically distinguished with regard to the underlying network topology. To this end, federated learning (FL) refers to a star topology where an orchestrating parameter server receives model updates from the actors, aggregates the updates, and broadcasts the aggregate. Decentralized learning (DL) constitutes arbitrary network topologies without an orchestrator, i.e., actors in the network learn by exchanging model updates within their neighborhood [2]. Actors within DL are typically referred to as nodes.

There are inherent risks and limitations with FL, such as that it may be challenging to find a trustworthy third party due to regulations or the desire for autonomy (e.g. for hospitals, banks, or other big corporations). Further, FL scales poorly in the number of nodes due to the communication bottleneck and the server constitutes a single-point-of-failure [2]. This has motivated research on fully decentralized systems, which eliminate the need for a central server. Instead, model parameters are directly communicated between peers in the learning setup using a communication protocol, such as gossip learning [3]. However, this approach is not well-suited for non-iid settings, where multiple distinct learning objectives may be present. In such cases, node selection during training is crucial for achieving efficient and effective learning.

The idea of each node identifying useful peers in the network to train a personalized model was proposed in [4]. Therein, nodes jointly learn a collaboration graph, via an alternating optimization method, that dictates whom to communicate to. A score-based method, decentralized adaptive clustering (DAC), was presented in [5] where each node scores its neighboring peers based on the the empirical loss, obtained by evaluating the received model parameters on the local dataset. While DAC manages to find beneficial nodes and identifies heterogeneous clusters in the network, model parameters from the nodes' training updates are still communicated over the network in plain text and the peers receiving the updates must hence be trusted. As such, DAC is vulnerable to inference attacks. This raises the question of how to ensure the privacy of the model parameters in decentralized machine learning systems. In many privacy-critical applications, differential privacy [6] is used in conjunction with FL to protect the data of nodes. Although this adds a layer of privacy, it comes at the expense of a deterioration in model performance.

In this work, we overcome this problem and introduce a communication-efficient and privacy-preserving algorithm named **Private Personalized Decentralized Learning (PPDL)**. We use multi-armed bandits to find beneficial collaborators and secure aggregation [7, 8] to hide individual updates. Our method works in a serverless decentralized setting, but can also apply to standard FL. We protect against inference attacks by only observing aggregated models.In our proposed method, a peer only observes an aggregate of model parameters, which substantially lessens the risk of inference attacks as compared to previous works.

Since a node only receives an aggregate of the parameter updates of M nodes at a given point in time,

*Corresponding Author.

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

it cannot infer a score on the similarity of any one of the peers in the aggregate (as in DAC); such a score can only be computed for the aggregate. Instead, our solution exploits dependencies between different group selections and makes use of adversarial multi-armed bandit optimization to efficiently find the subsets of peers that are beneficial for collaboration. Our experimental evaluations demonstrate that our approach offers a competitive solution for personalized decentralized learning that preserves data privacy under covariate shift and label shift and efficiently finds the beneficial collaborators within the network. Our solution has a communication efficiency and performance similar to that of previous methods, but adds a higher level of privacy.

2 Decentralized learning by finding useful collaborations

Problem formulation. We consider several DL tasks over a network of K nodes, each with a *private* data distribution $\mathcal{D}_i$ over the inputs $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$. Each node $i \in [K]$ has a model f_i with parameters $w_i \in \mathbb{R}^d$ and a loss function $\ell(f_i(w_i; x), y) : \mathbb{R}^d \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Each note wants to minimize its expected loss over its data,

$$w_i^\star = \arg\min_{w_i \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}_i} \left[\ell(f_i(w_i; x), y)\right]. \quad (1)$$

A challenge is to find similar nodes to collaborate with, without sharing data. If the distributions are substantially dissimilar, collaboration may result in decreased performance compared to local training without collaboration. In situations where some of the other nodes in the network have similar local data distributions, it may be beneficial to collaborate towards the goal in (1) by means of exchanging and aggregating model parameters.

However, revealing details of node data may be difficult or impossible due to privacy reasons. To address this issue, we propose a method for identifying nodes with similar local datasets in a private manner. We assume the nodes communicate over a network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{1, \ldots, K\}$ are the nodes and $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ are the edges between the nodes. The neighborhood of node $i \in \mathcal{N}$ is denoted by $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}, j \in \mathcal{N}\}$. Like [5, 9], node i want to find a set of nodes $\mathcal{M}_i \subseteq \mathcal{N}_i$ to exchange models with. In each round, the learning proceeds as follows. First, each node $i \in \mathcal{N}$ selects a set $\mathcal{M}_i \subseteq \mathcal{N}_i$ to receive model updates from. Second, the nodes in $\mathcal{M}_i$ submit their local models securely to node i by using secure aggregation, e.g., [8]. Third, node i computes the aggregated model from the nodes in $\mathcal{M}_i$ and aggregates it with its local model after which local training is initiated using the updated model.

Privacy. Although FedAvg is commonly advertised as being private, recent results have demonstrated attacks able to recover training data from the models [10]. To protect the nodes from such attacks, we utilize secure aggregation to ensure that a node who queried multiple model parameters from a subset of its neighbors only get to observe an aggregate of those models. The design of secure aggregation schemes is outside of the scope of this work but may be achieved for arbitrary networks by using Shamir's secret sharing scheme [11] as demonstrated in [8]. For our purposes, we assume that a node i queries a set $\mathcal{M}_i^{(t)} \subseteq \mathcal{N}_i$ of size M in round $t \in [T]$ and observes only the aggregate $\bar{w}_i^{(t)} = \sum_{j\in\mathcal{M}_i^{(t)}} \beta_j w_j$ where $\beta_j \geq 0$ satisfy $\sum_{j\in\mathcal{M}_i^{(t)}} \beta_j = 1$. Consequently, node i is presented with $C_i = \binom{|\mathcal{N}_i|}{M}$ different groups of nodes to choose among where we assume $|\mathcal{N}_i| \geq M$ for all $i \in [N]$. For example, in a fully connected network consisting of $K = 100$ nodes and secure aggregation schemes where $M = 2$ and $M = 3$, we have 4,851 and 156,849 different groups, respectively.

Multi-armed bandits. We have a challenging group-selection problem with many groups and few rounds. A node can only evaluate a group by its local accuracy, which is stochastic and non-stationary due to other nodes' actions. We use an online learning approach and model the problem for each node as an adversarial multi-armed bandit with C_i arms and T rounds [12].

The performance of a bandit algorithm is measured by pseudo-regret, which compares the expected rewards of the best arm and the algorithm. For adversarial bandits, the pseudo-regret per round decreases as $\mathcal{O}(\sqrt{C_i/T})$ [13]. This means a large C_i, as in our case, an algorithm cannot be expected to perform well in a few rounds. However, this assumes independent rewards; if rewards are dependent, pulling an arm can give information about other arms and reduce exploration [14].

In our problem, some groups share nodes. The number of groups that share u nodes with a given group is $\binom{M}{u}\binom{N-M-1}{M-u}$. For example, in a fully connected network with $N = 100$ and $M = 3$, there are 13,680 and 288 groups that share one and two nodes, respectively, with a given group. So, selecting one group out of the 156,849 could inform about 13,968 groups. To leverage this idea, we use of pseudo-rewards, as presented in [14].

Let the different groups available to node i be indexed from $1, \ldots, C_i$ and, w.l.o.g., let the reward from choosing group $j \in [C_i]$ at time t satisfy $r_j^{(t)} \in [0, 1]$. We define the pseudo-rewards $s_{l,j}^{(t)}(\alpha_j^{(t)}) \in [0, 1]$ as an upper bound on the expected reward on $r_l^{(t)}$ given that we observe $r_j^{(t)}$ for $j \in [C_i]$ and $l \in [C_i] \setminus \{j\}$. This is mathematically represented as:

$$\mathbb{E}\left[r_l^{(t)} | r_j^{(t)} = \alpha_j^{(t)}\right] \leq s_{l,j}^{(t)}(\alpha_j^{(t)}). \quad (2)$$

For $j = l$, we let $s_{j,j}^{(t)} = r_j^{(t)}$. Note that setting $s_{l,j}^{(t)}(\alpha_j^{(t)}) = 1$ for all $j, l \in [C_i]$, $l \neq j$ and $t \in [T]$, results in recovering the uncorrelated multi-armed bandit setting. Note that the inequality in (2) must be satisfied in order to achieve zero-regret asymptotically in the

number of rounds [14]. However, as our objective is to simply identify nodes with similar local data distributions within a fixed number of training rounds, the choice of pseudo-reward in (2) will mainly serve to trade-off between exploitation and exploration.

To use the correlated bandit framework in our setting, we notice that groups with large overlap have many parameters in common in the aggregation step, hence, it seems plausible that also their expected rewards should be closer than groups with less overlap. Therefore, we design the pseudo-rewards between two groups to be decreasing in the number of overlapping nodes. Furthermore, it is expected that the discrepancy in accuracy between groups with large overlap decreases over time, hence the time dependency in (2). Let $u_{l,j} \in \{0, \dots, M-1\}$ denote the number of overlapping nodes between group l and group j. We consider pseudo-rewards of the form

$$s_{l,j}^{(t)}(\alpha_j^{(t)}) = \min\left\{\alpha_j^{(t)} + \frac{q(t)}{u_{l,j}}, 1\right\} \tag{3}$$

where $q : [T] \to \mathbb{R}_+$ is a non-increasing function in time, i.e., $q(t_2) \leq q(t_1)$ for $t_2 > t_1$. We make this choice as the variance between node models is anticipated to decrease as models converge.

2.1 Private multi-armed bandits for node selection

In this section, we present our bandit algorithm for a node. For ease of notation, we omit the node index. Let $k^{(t)} \in [C_i]$ be the group chosen at time t and let $n_{k^{(t)}}(t)$ be the number of times it has been chosen. The reward from choosing group $j \in [C_i]$ is defined as $\mu_j(t) = \frac{\sum_{\tau=1}^{t} \mathbf{1}\{k^{(\tau)}=j\} r_j^{(\tau)}}{n_j(t)}$ and the pseudo-reward for group $l \in [C_i] \setminus \{j\}$ when group $j \in [C_i]$ is selected, is given by $\phi_{l,j}(t) = \frac{\sum_{\tau=1}^{t} \mathbf{1}\{k^{(\tau)}=j\} s_{l,j}^{(\tau)}(r_j^{(\tau)})}{n_j(t)}$. We reduce the problem size by selecting only competitive arms, i.e., arms whose minimum pseudo-rewards are higher than the maximum reward. To this end, we define the set of significant arms as $\mathcal{S}_i^{(t)} = \{j \in [C_i] : n_j(t) > t/N\}$ and let $\bar{k}^{(t)} = \arg\max_{l \in \mathcal{S}_i^{(t)}} \mu_l(t)$. The set of empirically competitive arms is defined as

$$\mathcal{A}_i^{(t)} = \left\{ j \in [C_i] : \min_{l \in \mathcal{S}_i^{(t)}} \phi_{j,l}(t) \geq \mu_{\bar{k}^{(t)}}(t) \right\} \cup \{\bar{k}^{(t)}\}. \tag{4}$$

Note that $\mathcal{A}_i^{(t)}$ is not monotonically decreasing in t as arms may be non-competitive in one round and competitive in the next. Once $\mathcal{A}_i^{(t)}$ has been obtained, an arbitrary multi-armed bandit algorithm may be applied over the set of arms. As we consider adversarial rewards, we employ the *Tsallis-Inf* algorithm that is known to achieve a pseudo-regret with the optimal scaling [15], where large $q(t)$ encourages exploration whereas a small $q(t)$ encourages exploitation.

3 Experiments

Our code is made available upon publication to encourage reproducibility [1]. All experiments were carried out on an Nvidia 3090 Ti GPU. We conduct experiments on various cluster configurations and employ the CIFAR-10 and Fashion-MNIST datasets, which are commonly used in the literature for decentralized machine learning evaluations on covariate and label shift, see Section 3.1 [16]. We follow previous work [5, 9] and assume a fully connected graph among the nodes. Our algorithm aims to find a sub-graph for each node that maximizes its local task performance. In other words, we want to find the best collaborators for each node based on its local, private data.

Baselines. In all experiments we use decentralized adaptive clustering (**DAC**) [5] as a baseline for comparison, as it is most similar to our work. In addition, we also make comparisons to a random gossip communication protocol (denoted **Random**) and an oracle (denoted **Oracle**) that has perfect information of cluster assignments and only communicates (randomly) within these. Moreover, we also compare with local training on the nodes where no communication is allowed (denoted **Local**).

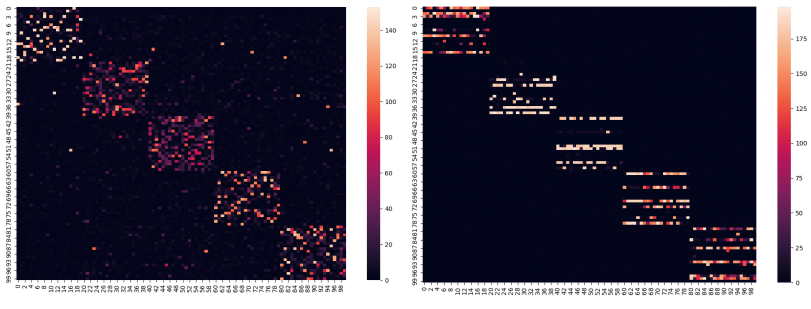
Covariate shift. To evaluate the performance of our method under non-iid data distributions, we replicate some of the experiments outlined in [5] for covariate shift with 100 nodes by dividing the data uniformly into four partitions, each with images rotated $0°, 90°, 180°$ and $270°$, respectively. We also experiment with heterogeneous cluster sizes by dividing the data into clusters of $0°, 180°, 350°$ and $10°$ rotation, with $70, 20, 5$ and 5 nodes in each cluster, respectively.

Label shift. Moreover, we also conduct experiments on label shift. As in [5], for the CIFAR-10 dataset we divide the data into two clusters based on labels, one for animal images and one for vehicle images. Additionally, we extend the experiments on label shift where we partition the data such that each node only has two labels, and these labels are grouped into clusters of five, where each cluster contains 20 nodes with the same two labels.

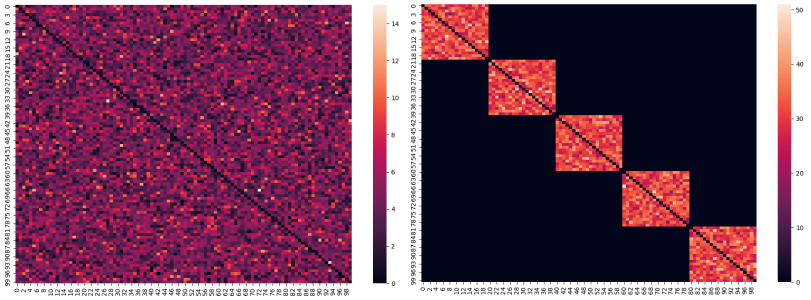
In our experiments, we evaluate all models on a test set with the same distributional shift as the training set for each node in the network. This is because the goal is to solve the local learning task for each node as effectively as possible. We use early stopping locally on each node.

Model and data. We use the same CNN architecture as [5], with three convolutional and two fully connected layers. We simulate 100 nodes for CIFAR-10 and Fashion-MNIST, and average results over three runs. Each node has equal data samples, uses the Adam optimizer and batch size of 8, and samples $M = 3$ other nodes per round. We train for three local epochs and 200 rounds. We use two $q(t)$ in (3): constant (PPDL) and exponentially decaying (PPDL-var), tuned by validation. We also tune learning rates using a validation set.

[1] https://github.com/edvinli/ppdl



(a) PPDL **(b)** DAC



(c) Random **(d)** Oracle

Figure 1. Heatmaps visualising how often node x communicates with node y for the four different methods on the CIFAR-10 dataset with 5 clusters.

Table 1. CIFAR-10 label shift test accuracy with 5 clusters.

Method	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Mean
PPDL	74.70	68.51	73.78	77.74	74.60	73.87
PPDL-var	71.82	80.40	78.72	82.06	76.05	77.81
DAC	**79.41**	**76.83**	**78.52**	**80.58**	**76.94**	**78.46**
Random	68.90	63.31	66.70	69.98	69.92	67.76
Local	77.11	88.79	82.60	61.83	68.32	75.73
Oracle	88.65	91.25	84.69	81.54	79.62	85.15

Table 2. Test accuracies for covariate shift on CIFAR-10 and Fashion-MNIST, with varying node numbers per cluster (70,20,5,5). Mean values over clusters are also provided.

	CIFAR-10				
Method	0°	180°	350°	10°	Mean
PPDL	52.37	45.21	50.60	51.03	49.80
PPDL-var	**54.63**	**47.27**	**51.84**	**53.30**	**51.76**
DAC	53.70	47.73	52.84	51.35	51.41
Random	54.85	44.70	52.64	52.43	51.16
Local	34.06	31.64	29.92	32.68	31.91
Oracle	55.04	46.80	38.35	38.00	44.55
	Fashion-MNIST				
PPDL	**84.62**	**81.81**	**81.01**	**82.11**	**82.39**
PPDL-var	80.68	81.12	80.42	80.66	80.72
DAC	82.48	80.44	79.85	80.43	80.80
Random	84.26	79.61	78.42	78.99	80.32
Local	78.72	76.83	77.40	77.26	77.55
Oracle	83.00	81.93	79.01	79.76	80.93

Table 3. Test accuracies for covariate shift on CIFAR-10 and Fashion-MNIST, with the same number of nodes per cluster (25). Mean values over clusters are also provided.

	CIFAR-10				
Method	0°	90°	180°	270°	Mean
PPDL	43.48	43.31	43.73	43.10	43.40
PPDL-var	45.06	44.05	44.60	43.14	44.22
DAC	**45.21**	**45.08**	**45.18**	**45.78**	**45.31**
Random	41.35	41.19	42.39	41.46	41.60
Local	32.01	32.34	31.47	33.07	32.22
Oracle	49.47	49.66	49.57	48.43	49.28
	Fashion-MNIST				
PPDL	80.69	81.12	80.43	80.66	80.73
PPDL-var	**80.81**	**81.71**	**82.36**	**80.19**	**81.26**
DAC	78.83	79.51	78.69	79.02	79.01
Random	80.20	80.72	79.3	79.99	80.05
Local	78.84	79.36	79.98	77.04	78.81
Oracle	82.86	83.18	84.25	83.79	83.52

Table 4. CIFAR-10 label shift test accuracy with 'animal' and 'vehicle' clusters.

Method	Vehicles	Animals	Mean
PPDL	51.86	36.31	43.81
PPDL-var	**52.86**	**36.33**	**44.60**
DAC	52.78	33.87	43.32
Random	44.79	30.00	37.40
Local	51.10	35.11	43.11
Oracle	57.17	39.74	48.45

Covariate shift. Tables 2 and 3 show the results of our covariate shift experiments with two cluster setups. Our method, PPDL, performs similarly to DAC, but with secure aggregation for privacy. Random favors large clusters and penalizes small ones, as seen in Table 2. DAC and PPDL avoid collaborating with "poisonous" nodes by their sampling schemes, improving test accuracy in the 180° cluster. Oracle has low test accuracies for small clusters, likely due to limited data (only 5 nodes per cluster). For the smallest clusters, 350° and 10°, PPDL and DAC find similar nodes in the large 0° cluster, improving performance. Thus, DAC and PPDL increase performance and fairness for smaller clusters that differ from large ones. The Fashion-MNIST results are less different between methods, likely due to the easier problem than CIFAR-10. Also, rotating images may not be challenging for small CNNs, as they can learn rotation-invariant representations with enough data. We analyze harder label shift problems ne

Label shift. The results of our label shift experiment with two clusters (animals and vehicles) are presented in Table 4. We observe that both PPDL and DAC perform well, with PPDL achieving superior results. The highest accuracy is achieved with PPDL-var, in which $q(t)$ is exponentially decayed. We note that Random performs worse than local training without collaboration, likely due to model poisoning caused by nodes communicating with incorrect clusters. For Random, the node models learn different representations for the different clusters, and when merging models from two distinct clusters, the resulting model is inferior due to the significant dissimilarity between the models, a phenomenon known as *client drift*. Both DAC and PPDL are able to mitigate this problem by identifying useful collaborators.

The results of our five-cluster experiment on CIFAR-10 are presented in Table 1, where each cluster consists of two unique labels. We observe that Random performs worse than the Local baseline on average also in this setting. Our experiments also reveal a high degree of variance within a cluster for the Local baseline, which can be attributed to the small size of node data. In contrast, the PPDL and DAC methods perform comparably and are able to correctly identify beneficial collaborators, as depicted in Figure 1.

4 Related work

Decentralized learning. Previous studies have demonstrated the effectiveness of gossip algorithms, as highlighted in references such as [3, 17, 18]. Furthermore, collaborative gossip algorithms, where nodes possess distinct local tasks, have been investigated in the context of multi-task learning (MTL) as seen in [4, 19]. While gossip learning has been demonstrated to be effective in convex optimization, its application in non-convex optimization, which is required for training deep neural networks, has not been as extensively studied. One of the first works that explored the use of gossip-based optimization for non-convex deep learning was conducted on convolutional neural networks (CNNs) in [20]. The authors demonstrated that high accuracies could be achieved at low communication costs using a decentralized and asynchronous framework. However, it is important to note that gossip learning is not well-suited for non-iid settings, where several distinct learning objectives may be present. Indeed, a protocol based on random communication between nodes does not take into consideration the benefits of node selection during training.

In centralized FL, methods based on hard clustering [21–23] can efficiently identify node clusters, but they limit the collaboration of nodes to their own clusters. This prevents nodes from utilizing useful information from similar clusters in forming a global model. Recent works have advanced decentralized learning of deep neural networks on non-iid data. [9] used expectation-maximization, while [24] improved node selection and communication cost with gradient-based cosine similarity and model pruning. [25] identified similar nodes by empirical loss, but only allowed hard clustering. [5] proposed a decentralized adaptive clustering algorithm that used empirical loss similarity to discover beneficial peers, but without privacy guarantees for model weights. This probability vector is then used for sampling similar nodes in the next communication round for each node, allowing for soft cluster assignments and communication within the entire graph. Empirical results demonstrate the effectiveness of this method in identifying clusters of nodes and improving the performance of the models. Although this method identifies useful node collaborations, there are a lot of privacy risks as model weights are being shared without any privacy guarantees.

Secure aggregation. Secure aggregation is a method to enhance node privacy in FL by protecting against server inference attacks [7, 26]. The idea relies on random masking of the node models, before uploaded to the server, such that the masks cancel out when models are aggregated. Extensions based on secret sharing schemes [11] have been proposed, e.g., [27]. For decentralized learning, where the communication topology may be arbitrary, only few works have considered privacy. One protocol for secure aggregation over arbitrary networks is presented in [8]. Specifically, for node i, the scheme consists of two phases: i) node $j \in \mathcal{N}_i$ broad-

casts a public key that is used to privately collect shares of a random mask generated by node $l \in \mathcal{N}_i$, ii) node i receives the masked models and the aggregated shares of the random masks at from each node and reconstructs the aggregated masks to recover the aggregated model. Note that all of the above schemes require the models to be mapped to a finite field, an operation that may impact the training. A step towards avoiding this step for secure aggregation over connected graphs was recently proposed in [28].

Multi-armed bandits. Random node sampling in FL and DL can be improved by biasing towards nodes' local losses [29]. Multi-armed bandits for node selection were introduced in [30] with rewards based on node latency and objective to minimize training time. Extensions for model averaging [31], asynchronous FL [32], and dropout and fairness handling [33] were proposed. However, multi-armed bandits may perform poorly when the number of arms is large or dependent. Dependency-based clustering [34] and pseudo-reward shrinking [14] are two methods to exploit dependencies and reduce the number of arms.

5 Conclusions and future work

We introduce **Private Personalized Decentralized Learning** (**PPDL**), a novel privacy-preserving node selection approach for personalized decentralized deep learning based on adversarial multi-armed bandits. Our approach uses secure aggregation to hide individual node metrics and exploits node dependencies to sample groups of collaborators efficiently. To the best of our knowledge, this is the first privacy-preserving node selection scheme for decentralized learning. We show that **PPDL** achieves comparable performance to existing (non-private) techniques on multiple experiments, while also providing privacy protection with secure aggregation.

For future research, it would be interesting to explore aggregation methods for models trained on different datasets in order to enhance the robustness of nodes to merging with other clusters. Another direction is to understand how privacy is affected by the number of nodes participating in the secure aggregation. Intuitively, as shown in, e.g., Section V.A in [8], privacy improves with larger group sizes.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Vol. 54. 2017, pp. 1273–1282.

[2] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. "Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[3] D. Kempe, A. Dobra, and J. Gehrke. "Gossip-based computation of aggregate information". In: *44th Annual IEEE Symposium on Foundations of Computer Science*. 2003, pp. 482–491.

[4] V. Zantedeschi, A. Bellet, and M. Tommasi. "Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. 2020, pp. 864–874.

[5] E. Listo Zec, E. Ekblom, M. Willbo, O. Mogren, and S. Girdzijauskas. "Decentralized adaptive clustering of deep nets is beneficial for client collaboration". In: *FL-IJCAI'22: International Workshop on Trustworthy Federated Learning* (2022).

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[7] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. "Practical secure aggregation for privacy-preserving machine learning". In: *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. 2017.

[8] K. Tjell and R. Wisniewski. "Private Aggregation With Application to Distributed Optimization". In: *IEEE Control Systems Letters* 5.5 (2021), pp. 1591–1596.

[9] Y. Sui, J. Wen, Y. Lau, B. L. Ross, and J. C. Cresswell. *Find Your Friends: Personalized Federated Learning with the Right Collaborators*. 2022.

[10] D. I. Dimitrov, M. Balunovic, N. Konstantinov, and M. Vechev. "Data Leakage in Federated Averaging". In: *Transactions on Machine Learning Research* (2022).

[11] A. Shamir. "How to Share a Secret". In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613.

[12] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "The Nonstochastic Multiarmed Bandit Problem". In: *SIAM Journal on Computing* 32.1 (2002), pp. 48–77.

[13] J.-Y. Audibert, S. Bubeck, et al. "Minimax Policies for Adversarial and Stochastic Bandits." In: *COLT*. Vol. 7. 2009, pp. 1–122.

[14] S. Gupta, S. Chaudhari, G. Joshi, and O. Yagan. "Multi-Armed Bandits With Correlated Arms". In: *IEEE Transactions on Information Theory* 67.10 (2021), pp. 6711–6732.

[15] J. Zimmert and Y. Seldin. "Tsallis-INF: An Optimal Algorithm for Stochastic and Adversarial Bandits". In: *J. Mach. Learn. Res.* 22.1 (July 2022).

[16] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. *Advances and Open Problems in Federated Learning*. 2019.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. "Randomized gossip algorithms". In: *IEEE transactions on information theory* 52.6 (2006), pp. 2508–2530.

[18] R. Ormándi, I. Hegedűs, and M. Jelasity. "Gossip learning with linear models on fully distributed data". In: *Concurrency and Computation: Practice and Experience* 25.4 (2013), pp. 556–571.

[19] P. Vanhaesebrouck, A. Bellet, and M. Tommasi. "Decentralized Collaborative Learning of Personalized Models over Networks". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. 2017.

[20] M. Blot, D. Picard, M. Cord, and N. Thome. "Gossip training for deep learning". In: *arXiv preprint arXiv:1611.09726* (2016).

[21] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran. "An efficient framework for clustered federated learning". In: *Advances in Neural Information Processing Systems* 33 (2020).

[22] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh. *Three approaches for personalization with applications to federated learning*. 2020.

[23] F. Sattler, K.-R. Müller, and W. Samek. "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints". In: *IEEE transactions on neural networks and learning systems* 32.8 (2020), pp. 3710–3722.

[24] Z. Ma, Y. Xu, H. Xu, J. Liu, and Y. Xue. "Like Attracts Like: Personalized Federated Learning in Decentralized Edge Computing". In: *IEEE Transactions on Mobile Computing* (2022).

[25] N. Onoszko, G. Karlsson, O. Mogren, and E. L. Zec. "Decentralized federated learning of deep neural networks on non-iid data". In: *2021 ICML Workshop on Federated Learning for User Privacy and Data Confidentiality* (2021).

[26] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova. "Secure Single-Server Aggregation with (Poly)Logarithmic Overhead". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1253–1269.

[27] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E. Ali, B. Guler, and S. Avestimehr. *LightSecAgg: a Lightweight and Versatile Design for Secure Aggregation in Federated Learning*. 2021.

[28] K. Tjell and R. Wisniewski. *Privacy in Distributed Computations based on Real Number Secret Sharing*. 2021.

[29] Y. Jee Cho, J. Wang, and G. Joshi. "Towards Understanding Biased Client Selection in Federated Learning". In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. 2022.

[30] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu. "Multi-Armed Bandit-Based Client Scheduling for Federated Learning". In: *IEEE Transactions on Wireless Communications* 19.11 (2020), pp. 7108–7123.

[31] T. Kim, S. Bae, J.-w. Lee, and S. Yun. *Accurate and Fast Federated Learning via Combinatorial Multi-Armed Bandits*. 2020.

[32] H. Zhu, J. Kuang, M. Yang, and H. Qian. "Client Selection with Staleness Compensation in Asynchronous Federated Learning". In: *IEEE Transactions on Vehicular Technology* (2022).

[33] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya. "Stochastic Client Selection for Federated Learning With Volatile Clients". In: *IEEE Internet of Things Journal* 9.20 (2022), pp. 20055–20070.

[34] R. Singh, F. Liu, Y. Sun, and N. Shroff. *Multi-Armed Bandits with Dependent Arms*. 2020.