FedCD: Personalized Federated Learning via Collaborative Distillation

Sabtain Ahmad
Institute of Information Systems Engineering
Vienna University of Technology
Vienna, Austria
sabtain.ahmad@tuwien.ac.at

Atakan Aral
Faculty of Computer Science
University of Vienna
Vienna, Austria
atakan.aral@univie.ac.at

***Abstract*—Federated learning enables the creation of a centralized global model by aggregating updates from the locally trained models from multiple clients. While powerful, such an architecture is limited to applications where the needs of heterogeneous clients can be served by a single global model. It does not cater to the scenarios where each client independently designs its own model. Task and data heterogeneity inherent to such scenarios demand each client to specialize in the local setting while still being able to collaborate and transfer the acquired knowledge to the rest of the federation without sharing the data or the model. In this work, we utilize ensemble and collaborative learning to design a framework that enables the training of personalized models for heterogeneous clients with different learning capacities using federated learning. Empirical evaluations performed on the CIFAR100 dataset demonstrate that our framework is able to consistently improve the performance of all the participating models and outperform the independently trained models on the complete training set without collaboration. We analyze that all participants benefit from collaborative distillation and boast an average 1.4% increase in performance. Moreover, a comparison with the state-of-the-art approaches demonstrates that our framework outperforms the Federated Learning and Federation Distillation methods by up to a $2\times$ increase in the average global accuracy.**

***Index Terms*—Federated learning, knowledge distillation, collaborative learning, model heterogeneity, personalized learning.**

I. Introduction

Federated learning (FL) has garnered a lot of attention with its remarkable ability to collaboratively train a global model while preserving the privacy of the participating clients [1]. FL runs on the computing continuum as it performs parts of its computations on user/edge devices and the rest on the cloud server. Training a global model involves each client performing a few rounds of local training using its local data after receiving the copy of the global model and then compiling and sending the local updates to the parameter server, where updates from the clients are aggregated to update the global model. This iterative process continues until convergence, or the stopping criterion is met. The periodic exchange of parameters between clients and the server can quickly become an overhead as the ever-increasing complexity of modern deep neural networks (DNNs) results in a massive number of parameters, sometimes reaching up to 96 MB [2].

One of the many challenges faced by FL entails heterogeneity that appears in all aspects of the learning process [3]. Each participating client is assumed to possess different networking and computational power, which results in system heterogeneity. This may render a number of devices unable to train a previously agreed-upon architecture of a centralized model. The asynchronous scheme of FL and active sampling introduced in [4][5] can be used to work around this problem; however, that still requires excluding the resource-constrained devices which do not meet the criteria.

The second kind of heterogeneity that comes up quite often is referred to as data heterogeneity or formally known as the non independent and identically distributed (non-IID) data problem [6]. The problem of non-IID data distributions appears because of the fact that the local data on each client are not sampled from global joint distribution, consequently leading each client to have a different objective as they may not share common minimizers. This problem of client-drift not just degrades the performance of the global model but may also result in a higher number of communication rounds [6]. In light of the client drift problem, many approaches have been proposed mainly from two perspectives. 1) Use of additional (proxy) data information to address the model drift issue caused by non-IID data. FedDF [7] utilizes proxy data in the server to enable ensemble distillation. 2) Learning a generative network to augment the training process with the synthetic data as done in [8]. Though the global model is improved, the inherent heterogeneity among the local models is not fully addressed by only refining the global model.

In addition to data heterogeneity, we attend to a different type of heterogeneity that concerns each client requiring an individual model. Vanilla FL (FedAvg) [1] works under the assumption that all clients agree on the particular architecture of a global model. This simple yet critical assumption makes the collaboration of hundreds of devices possible. However, many applications, particularly in business-facing settings, could benefit from each client having their own unique model well suited to a particular client's data and resources. For example, in cross-silo applications where several organizations collaborate without sharing their private data, it may be more desirable and beneficial for them to train their personalized model that fits their needs and distinct specifications. Another

reason why owning a private model may be more appropriate is privacy and intellectual property concerns. Furthermore, AI as a service model demands more specialization. Imagine a typical AI vendor of, e.g., virtual shopping assistants that may have dozens of clients. Each client model is distinct and solves a peculiar task. In a conventional setting, each client model is trained only on the data at its disposal. It would be highly beneficial if clients trained on their local data could transfer their acquired knowledge to other clients without sharing their private data or model. Each client having their own model can also prove pivotal for addressing the non-IID data problem of FL since a natural way to tackle statistical heterogeneity is to have personalized models for each user. Apart from the non-IID data problem, another issue faced by the FL is the communication overhead which prohibits the use of large-sized models as it is proportional to the model sizes [9].

In this work, in order to work around the above-mentioned issues, we employ collaborative learning based on knowledge distillation to introduce a communication-efficient framework that facilitates clients with different capacities to train localized models. The motivation for our work stems from the recognition that traditional FL involving a single shared model struggles to adapt to all clients, especially when the data across clients are non-IID. The client-drift issue may render the global model irrelevant for some users. Thus, a single global model is inadequate for most practical applications [10]. Moreover, users may have the incentive to participate in FL only if the global model is far more accurate than the local model they can train in isolation on their own data. We look at FL from the viewpoint of an individual user and ask the question; how can clients individually improve their local models and help improve the rest of the users' models?

Answering this question requires a way of assembling knowledge from the local models and transferring this information to every user without overriding their local characteristics. Various challenges involved in this work include; 1) enabling knowledge distillation between a number of clients without requiring a large pre-trained teacher network. 2) reducing the communication overhead between the clients and the server. 3) improving the accuracy of all the participants through collective knowledge sharing. To overcome the first two obstacles, we employ collaborative learning-based knowledge distillation (discussed in Section III). Then, to handle the problem of generating soft target supervision that can boost the performance of all users with different learning capacities, we tend to ensemble learning which yields better results when diversity is present among the output of users [11].

We propose a novel personalized federated learning framework based on collaborative distillation (FedCD). To summarize, Our main contributions are listed as follows:

- C1 A new federated learning framework of knowledge distillation based on collaborative learning is proposed, letting users design models independently.
- C2 A communication efficient scheme to share local model updates with the parameter server based on online knowledge distillation is introduced.
- C3 Parameter server is relieved from the burden of training a complicated global model as we introduce a low-complexity ensemble method to dynamically generate high-quality soft targets for online knowledge distillation.

The remaining of this paper is organized as follows. In Section II, we touch upon the related work. Section III presents the concepts which constitute the background. Section IV shows the architecture overview and the principle of the proposed FedCD. In Section V, we evaluate the performance of FedCD and compare it to the state-of-the-art solutions. Section VI concludes the paper.

II. Related Work

A plethora of research has followed up since the introduction of FL as a privacy-preserving decentralized machine learning paradigm. Subsequent work mostly tackles the challenges faced by FL, including data heterogeneity [12] [13] and reducing communication overhead [14] [15].

Knowledge distillation (KD) has emerged as an effective solution for tackling the non-IID data problem. Federated distillation fusion (FedDF) employs KD to improve the global model; however, requiring that an unlabeled dataset is available with samples from identical distributions [7]. In order to reduce the communication overhead, FedDistill uses KD to allow users to share local logits instead of the model parameters [16]. FedMD [17] uses public data and involves two training steps; in the first step, each user's model is trained on the public data and then on their private data and afterward uses knowledge distillation to communicate the output scores from public data to other users. While these approaches require auxiliary data, FedGen [8] performs KD in a data-free manner by training a generator network. These approaches have demonstrated their efficacy in improving the performance of the global model. However, the inherent data and system heterogeneity are not fully addressed by only improving the global model.

Our approach differs from these methods in the following aspects: 1) instead of every user agreeing upon a single global model, each user designs and trains their own personalized model well suited to their needs. 2) in order to reduce the communication overhead, each user shares their local logits instead of the model parameters. 3) an ensemble method to generate high-quality soft targets is introduced to enable knowledge distillation.

III. Background on Co-distillation

Knowledge distillation involves optimizing a student network under the supervision of the teacher network. More precisely, the loss is defined as the Kullback-Leibler (KL) divergence, which measures the discrepancy between two probability distributions, in this case, the difference between the soften output of the teacher network and student network as in Equation 1 [18].

$$L_{KD} = \frac{1}{n} \sum_{i=1}^{n} T^2 KL(p_i, q_i) \tag{1}$$

Here n is the batch size, T is the temperature used to make the target to be soft, providing more information about the classes similar to the predicted class, whereas p and q represent the soften probability distribution produced by the teacher and student network, respectively.

KD mandates transferring knowledge from a *pre-trained* teacher network to a student network. However, the online version of KD, often referred to as collaborative (or Co) distillation (CD), is free from such requirements and distills knowledge on the fly from teacher to student network [19]. Teacher knowledge can be constructed in various ways. For instance, it can be a pre-trained teacher model's logit or an ensemble of other student models' logits, which could yield better results than individual model predictions [16]. Leveraging this observation, multiple approaches have been proposed, such as Deep Mutual Learning (DML) [20] and KDCL [21] that fuse the training process into a simplified single-stage and treat all networks as students, consequently enabling knowledge transfer in both ways; from teacher to student and vice versa. KDCL regards each network as a student and effectively generates soft targets by combining the logits from every network. This ensures students with different learning capacities benefit from collaborative learning. The loss function is defined as the KL loss to distill the knowledge of the soft target to each student network.

$$L_{CD} = \frac{1}{n}\sum_{i=1}^{n} L_{CE_i} + \lambda L_{KD_i} \tag{2}$$

Here L_{CE} is the cross-entropy (CE) loss, L_{KD} is KL divergence between the student and the soft target, and λ is the tradeoff weight between L_{CE} and co-distillation loss [21].

IV. Federated Learning via Co-distillation

A. Problem Statement:

Assume n clients, each with their own private dataset $D_j := x_i^j, y_i^j$, which may or may not be drawn from the same distribution, and an independently designed deep neural network DNN_j. It is essential to mention that DNNs can have different architectures depending upon the specifications of each client. Furthermore, the weights or the hyperparameters need to be shared neither with the parameter server nor among the clients. The objective is to design a collaboration framework that fuses the training of all networks into a single step and distills the knowledge accumulated from all models to improve the performance of DNN_j beyond individual effort.

B. FedCD

In our framework, illustrated in Figure 1, there is no single global model; instead, each user has its own model, and all models are regarded as student models. Each user independently trains their model on their private data and uploads the logits to the parameter server, which is responsible for generating the supervision by combining the outputs of all the models. If there are n users, j^{th} user's logit is defined as z_j. The server generates the teacher logit z_t, as in Equation 3.

$$z_t = h(z_1, z_2, \ldots, z_n) \tag{3}$$

Algorithm 1 The FedCD framework for enabling localized federated learning.

Input:
Output:

1: **while** not stopping_criterion **do**
2: **procedure** LOCAL TRAINING( ) ▷ at each user
3: $z_t = downloadTeacherLogits()$
4: **for** m steps **do**
5: $B,\ y_B \leftarrow D_j$
6: $z_j,\ \hat{y_B} \leftarrow DNN_j(B)$
7: $loss_j \leftarrow L_{CE_j} + \lambda KL(z_j, z_t)$
8: **end for**
9: **Communicate** local logits (z_j) to the server.
10: **end procedure**
11: **procedure** GLOBAL ENSEMBLING( ) ▷ at the server
12: **for** each user j = 1,2,..., N **do**
13: $z_j \leftarrow getuserlogits(j)$
14: **end for**
15: $z_t \leftarrow h(z_1, z_2, ..., z_n)$
16: **end procedure**
17: **end while**

Here h is a function for generating high-quality teacher logit. We distill the knowledge of teacher logit to each model as defined in Equation 2. Henceforth, the question of how much knowledge is distilled to each user boils down to generating high-quality teacher logit. One approach involves just averaging the student logits together to find the teacher logit as accomplished in [17]. Even though the approach is quite simple, it requires treating all users equally regardless of how much loss they incur during training. We propose an effective yet uncomplicated ensemble method for generating teacher logit, which assigns weights to users depending on how much loss they incur on a validation set. Consequently, the problem of finding the best teacher logit is reformulated as finding the best linear combination of student logits as illustrated in Equation 4.

$$\min_{\gamma \in R^n} \quad L_{CE}(\gamma^T \mathbf{Z}, \mathbf{y}) \quad \text{s.t.} \quad \sum_{j=1}^{N} \gamma_j = 1, \gamma_j \geq 0 \tag{4}$$

Where $\mathbf{Z}$ is a matrix containing student logits as columns. This process is formally described in Algorithm 1. In lines 2-8, each client downloads teacher logits, performs a few steps of training their models on their local private data and computes the local logits z_j, which are uploaded to the server. Line 11 contains the method in which the server ensembles user logits to create teacher logits z_t. During the subsequent iterations, each user downloads the teacher logits and continues the training with the objective of minimizing both the CE and KL divergence between the teacher and local logits as described in Equation 4 until the stopping_criterion is met, i.e., the maximum number of global iterations or users' accuracy reaches the threshold value.

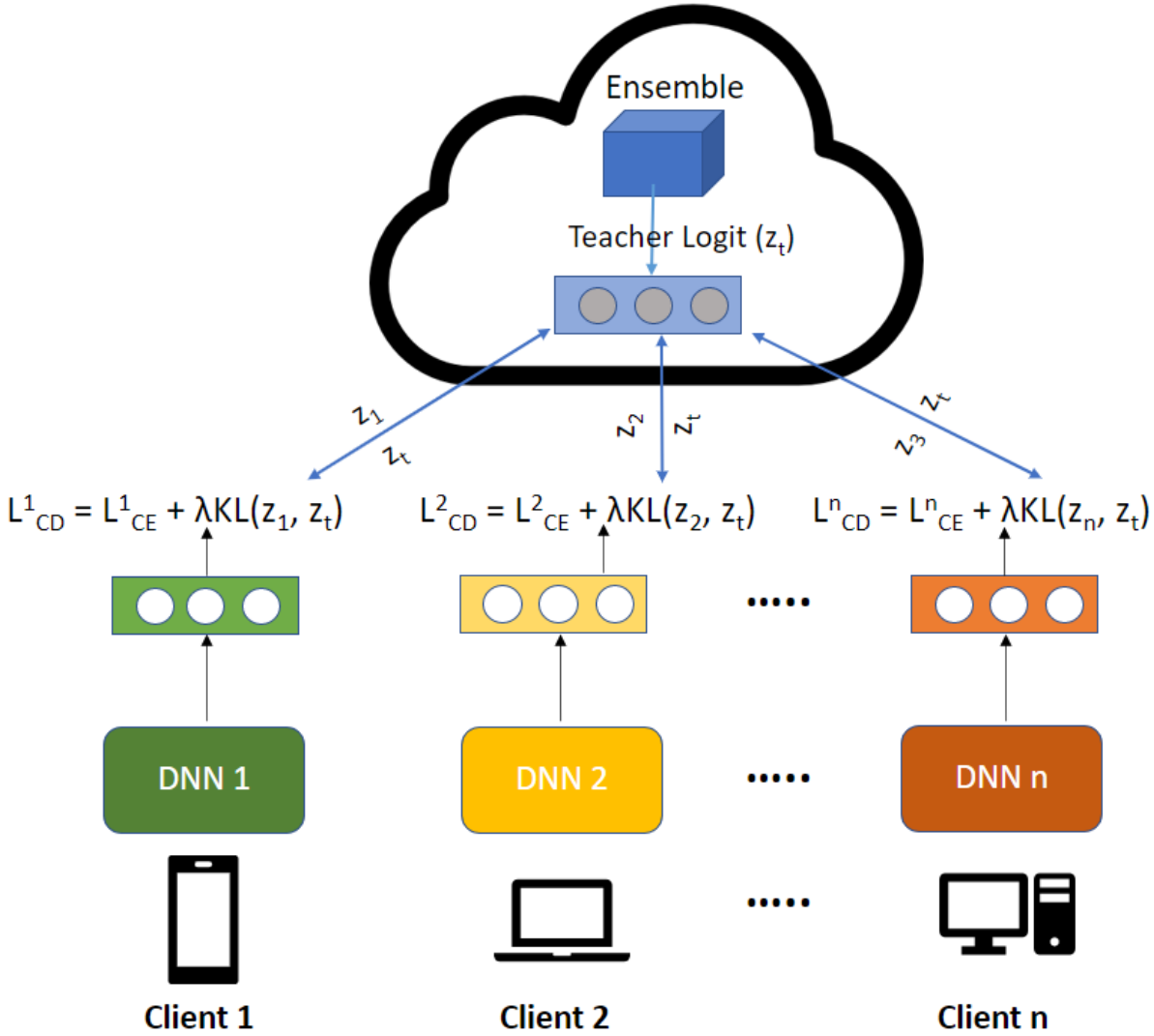


Fig. 1: Overview of the FedCD framework, which enables each client to train a personalized model. The server aggregates the local logits to generate the supervision to distill the knowledge to each client. The collaboration of clients equipped with personalized models helps improve performance.

V. Evaluation

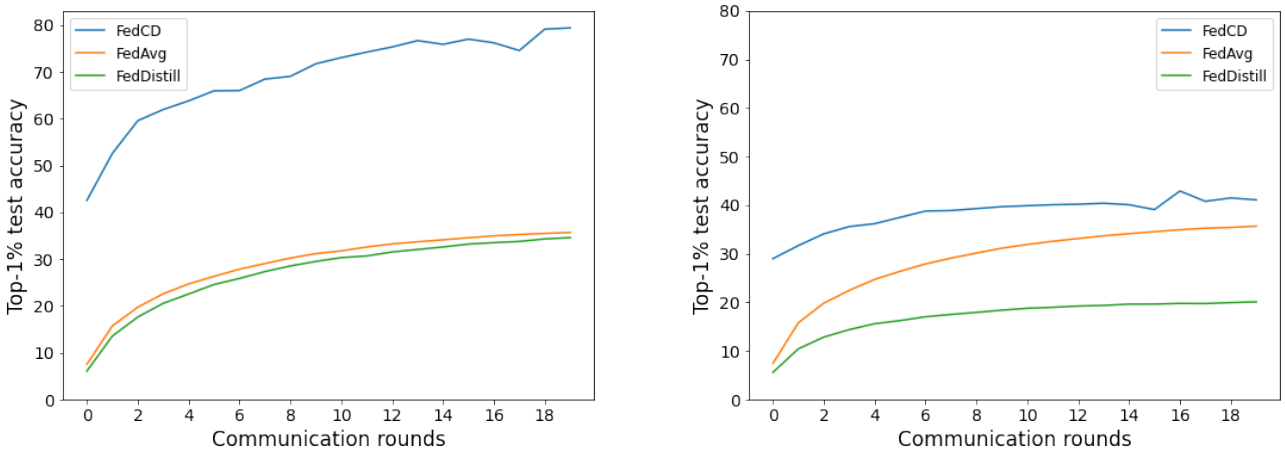
In this section, we perform a series of experiments to evaluate our localized federated learning framework (FedCD) on the benchmark dataset for computer vision, CIFAR100. The aim of the experiments is to study the effect of model and data heterogeneity and demonstrate empirically how a collaboration of users with different learning capacities can help enhance each other's performance. Therefore, these experiments involve models with different architectures of varying sizes and complexity. We compare the performance of our proposed approach to classical FL of FedAvg [1] and FedDistill, which is a data-free knowledge distillation-based approach that shares logit vectors instead of the model parameters [15].

A. Dataset

CIFAR-100 contains 100 classes consisting of 32 x 32 color images. It is comprised of a training and a test set with 50,000 and 10,000 images, respectively. We follow the approach in [8] to create IID and non-IID datasets. For the non-IID setting, we use Dirichlet distribution [22] in which a smaller α indicates higher data heterogeneity. We model non-IID data distributions by setting α to 0.5. For our ensemble method to generate teacher logit and measure the generalizability of students, we sample a validation set from the training set consisting of 5,000 images, 50 samples per class.

B. Experimental Setup

We follow the ResNet training procedure. We set the learning rate $\in \{0.001, 0.1\}$, weight decay to 0.0001, batch size to 64, and the number of global iterations to 200. The parameter settings for implementing baselines are consistent with their citation to achieve the best results. We use python alongside Pytorch (version 1.6.0) as a development environment. And we use a single GPU, NVIDIA T500, for hardware acceleration.



(a) IID

(b) non-IID

Fig. 2: Test accuracy of FedAvg, FedDistll, and FedCD on both IID (left) and non-IID data (right)).

TABLE I: The comparative result of different networks. FedCD improves the performance of all the networks beyond their individual capacity.

Model	Individual	Accuracy (%) w.r.t. the number of users				
		2	4	6	8	10
ResNet8	54.80	55.72	55.52	54.66	55.22	54.83
Resnet14	64.35	65.47	64.10	64.44	64,75	64.62
Resnet20	67.68	68.78	68.21	67.86	68.02	68.25
Resnet32	70.05	70.89	70.84	70.63	70.76	70.14
Resnet44	71.64	72.34	71.70	71.93	72.03	71.82
Resnet50	78.73	79.16	79.68	79.77	80.07	80.54
Resnet56	72.08	72.47	72.04	72.70	72.78	72.72
Resnet110	73.14	74.24	73.68	74.06	74.02	73.97
Resnet8x4	71.28	72.59	71.79	72.35	72.26	72.13
Resnet32x4	77.39	79.15	79.04	79.09	78.97	78.82

C. Results

In this section, we evaluate the performance of FedCD and baselines on the CIFAR100 dataset with IID and non-IID data distributions. We run 200 global communication rounds with 10 user models. First, we demonstrate the effectiveness of our localized algorithm where each of the 10 users chooses a unique ResNet model by comparing the average local test accuracy of each client as reported in Table I. It contains the accuracy of each personalized model trained in isolation on the complete training set and the performance of these models trained in collaboration with a varying number of users. We regard the independently trained models as baselines and compare them with the models trained in collaboration. We observe that the models trained in collaboration are able to benefit from the collective knowledge gathered from the rest of the users, acquire substantial improvements, and outperform the isolated trained models. We notice that all users benefit from knowledge distillation. This observation confirms our intuition that the knowledge distilled from the teacher logit composed by ensembling the logits from the participating users helps improve the performance of all models.

TABLE II: FedCD's average accuracy compared to baselines.

Settings	FedAvg	FedDistill	FedCD	FedCD_basic
IID	35.94	34.85	70.78	**73.95**
non-IID	35.88	20.11	42.92	**44.51**

Table II compares the performance of our proposed method with the baselines and demonstrates the superiority of our approach in both IID and non-IID settings. It is essential to mention that we use average test accuracy across users as a metric for comparison purposes since our approach focuses on personalized local models instead of training a single global model. Furthermore, for a fair comparison, we introduce FedCD_basic, which restricts each user to have the same model instead of allowing each user to train a different model since both the baselines FedAvg and FedDistill involve training one single global model. For both IID and non-IID data settings, we can observe from Table II that both FedCD_basic and FedCD outperform the baselines with a considerable margin (100 and 20-121%, respectively). In the case of IID, FedCD_basic and FedCD report an accuracy of 73.95 and 70.78, respectively which is an increase of almost 100% compared to the two baselines. This result confirms our motivations since the advantage of FedCD is induced from the knowledge distilled from teacher logit to local users. This knowledge is otherwise not accessible by FedAvg and FedDistill because as soon the parameter server is done aggregating and creating an update of the global model, the local updates from participating users are discarded.

The performance of both FedAvg and FedDistill deteriorates in the case of non-IID, whereas in comparison, FedCD is quite

TABLE III: Communication cost (upstream + downstream) incurred during training in IID settings.

Method	FedAvg	FedDistill	FedCD
Communication cost (MB)	37928.40	16.0	10.24

robust to non-IID data distributions. Although the FedAvg and FedDistill exhibit similar performance for IID settings, FedDistill experiences a non-negligible performance drop of 41% in the case of non-IID since it focuses on reducing the communication overhead; as highlighted in Table II. The FedCD_basic achieves the best performance of 44.51% which is about 24% higher than FedAvg and 121% higher than the FedDistil. The slight difference in the performance of FedCD compared to the FedCD_basic is due to the fact that each of the ten users, in the case of FedCD, is equipped with a model of varying complexity. This is consistent with the observation that a student network guided by a complex network yields superior performance [18]

In Figure 2, we present the learning curves from all the methods over the range of communication rounds. Fed_basic and FedCD demonstrate similar performance; therefore, we only present the results from FedCD. The results displayed in Figure 2 are generally consisted with the Table II. From the beginning, FedCD is able to constantly outperform FedAvg and FedDistill on CIFAR100 dataset in both IID and non-IID cases. We observe that our proposed method shows robustness at different communication rounds.

Finally, in order to evaluate the communication efficiency, we compare our proposed method with the baselines with respect to the communicated bits. We train for 200 iterations and measure the number of bits communicated both for upload and download. For each global iteration, FedCD and FedDistill exchange 64x100 and 100x100 logits for both uplink and downlink, respectively. On the other hand, FedAvg exchanges 23,705,252 trainable model parameters per iteration. Table III shows the amount of upstream and downstream communication required to achieve the best accuracy for each respective method. We observe that FedCD communicates only 10.24 MB of data, which is a reduction by a factor of 1.6x and 3700x compared to FedDistill and FedAvg, respectively.

VI. Conclusion

In this work, we propose FedCD as a personalized FL framework to improve the performance of FL in the presence of model and data heterogeneity. The proposed framework allows clients with different network and computation capabilities to design their own unique models and benefit from the knowledge shared by the rest of the users. In particular, FedCD trains a personalized model for each client; clients share their local logits, which are ensembled by the parameter server to create teacher logits which are then downloaded by each client to distill the knowledge from the peers. Empirical results demonstrate that our approach outperforms the classical FedAvg and knowledge distillation-based approaches in both IID and non-IID settings. In future work, we plan to explore different ensemble methods to generate high-quality teacher logits and study their effect on the performance and convergence of FedCD.

Acknowledgments

The work presented in this paper has been supported by the CHIST-ERA grant CHIST-ERA-19-CES-005 and by the Austrian Science Fund (FWF): I 5201-N.

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 *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: a compact task-agnostic BERT for resource-limited devices," in *Annual Meeting of the Association for Computational Linguistics*, 2020.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[4] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE International Conference on Communications*, 2019, pp. 1–7.

[5] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks," in *IEEE VTS Asia Pacific Wireless Communications Symposium*, 2019.

[6] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in Neural Inf. Proc. Systems*, vol. 33, pp. 7611–7623, 2020.

[7] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2351–2363, 2020.

[8] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *International Conference on Machine Learning*, 2021, pp. 12 878–12 889.

[9] A. Aral, M. Erol-Kantarci, and I. Brandić, "Staleness control for edge data analytics," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 2, 2020.

[10] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arXiv preprint arXiv:2002.04758*, 2020.

[11] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[12] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Int'l Conference on Machine Learning*, 2020, pp. 5132–5143.

[13] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[14] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," in *Workshop on ML on the Phone and other Consumer Devices*, 2019.

[15] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning," in *Workshop on Machine Learning on the Phone and other Consumer Devices*, 2018.

[16] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "Federated knowledge distillation," in *Machine Learning and Wireless Communications*, D. Gündüz, Y. C. Eldar, A. Goldsmith, and H. V. Poor, Eds. Cambridge University Press, 2022, ch. 16.

[17] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," in *WS on FL for Data Privacy and Confidentiality*, 2019.

[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS DL and Representation Learning Workshop*, 2015.

[19] R. Anil, G. Pereyra, A. T. Passos, R. Ormandi, G. Dahl, and G. Hinton, "Large scale distributed neural network training through online distillation," in *International Conference on Learning Representations*, 2018.

[20] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *IEEE Conf. on CV and PR*, 2018, pp. 4320–4328.

[21] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Luo, "Online knowledge distillation via collaborative learning," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 11 020–11 029.

[22] H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," in *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.