Online federated learning with imbalanced class distribution

KONSTANTINOS GIORGAS*, Athens University of Economics and Business, Greece
IRAKLIS VARLAMIS, Harokopio University of Athens, Greece

The federated learning paradigm can be a viable solution for handling huge datasets, and for taking advantage of powerful processing nodes on the edge. The process of online federated learning can be employed in order to maximise the potential of federated learning by re-training a shared model on the edge nodes and merging the updated models centrally. This approach allows edge nodes to exchange knowledge without exchanging their own training data, thus preserving their privacy. In this work, we examine the online federated learning approach in an extreme case of imbalanced class distribution between the central and the edge nodes. We examine the effects of different parameters of the online federated learning process and propose a technique that boosts the classification performance above that of the baseline centralised learning approach.

CCS Concepts: • **Computing methodologies → Machine learning**; *Parallel Computing Methodologies*; • **Information systems** → Information systems applications.

Additional Key Words and Phrases: federated learning, online learning, class imbalance

1 INTRODUCTION

Deep learning models become popular in the analysis of text, images, video, audio or other data streams and the extraction of high-level features, since they provide improved performance over the traditional machine learning models. The abundance of tools, platforms and specialized hardware components that allow faster development, training and execution of deep learning models, further increased their popularity [5]. The increased expectations from deep learning applications consequently raised the requirements in computational resources. On the other side, the popularity of Internet of Thing (IoT) devices created the need to process massive data quantities, efficiently and, in some case, without breaching users' privacy. This raised the need for deep learning on the edge or in a combination of edge and the cloud [7]. For example, in the automotive domain, the data which are collected from self-driven vehicles can be used to train models on the edge, but cannot necessarily be shared with other users or uploaded to the cloud. Moving deep learning computations to the edge may help in enhancing users' privacy [11] and for saving network bandwidth. It may also help to train large-scale deep learning models faster, taking advantage of the distributed computing principles and establishing the "*federated learning*" paradigm [13].

In federated learning, the model is trained across multiple decentralized nodes that hold their local data samples, without exchanging them. The nodes exchange the trained models and continue training in rounds. In this work we examine how the distribution of training samples affects the quality of the distributed trained model. We evaluate several training alternatives and their parameters and conclude that it is beneficial for the training process to re-train the shared model more frequently using fewer training samples in every round and consequently merge the distributed re-trained models more frequently. Using only the training samples that are "harder" for the model to classify correctly and not the whole training dataset each time, further improves the federated learning process performance. It further helps to tackle online learning issues such as catastrophic forgetting [10], which may arise from imbalanced training data and improper sampling.

The sections that follow briefly present the most recent works in online federated learning (Section 2) and the proposed approach and the various alternatives (Section 3). Section 4 contains a comparative evaluation of the alternatives and depicts the effect of various decisions and parameters to the overall performance. Finally, Section 5 concludes the article and presents our next steps.

2 RELATED WORK

According to concept of Federated Learning the training data are distributed across multiple devices, and training takes place locally for preventing data leakage. Handling the statistical variance problems of training data (e.g. training with non-IID data [10]), maintaining the privacy of data on the edge [11], improving the resilience and robustness of the federated models [1] and boosting the communication efficiency [3] and scalability [2] of the solutions are the major research topics in the area of federated machine learning [6].

Our work mainly emphasises on the statistical challenge of handling the discordance of training samples across edge nodes. Authors in [14] experimented with MNIST and CIFAR-10 datasets using CNNs. They created two extreme cases (each client either receives a single class only partition, or 2 partitions from 2 randomly selected classes) and employed the Federated Average algorithm [8] for model merging. Authors in [10] focused on the instability of the *FedAvg* algorithm on non-IID datasets. Using a simplified version of the 11-layer VGG11 neural network on the CIFAR dataset, and a 4-layer CNN on the speech commands dataset, they concluded that Top-k sparsification sampling is more stable in such setups. Finally, authors in [9] implemented a lifelong federated learning strategy, called Federated Curvature (FedCurv), which outperformed FedAvg, using a CNN on the MNIST dataset. They also experimented with the number of epochs in each round.

Our approach combines the merits of online learning with the simplicity of FedAvg, by sampling the training data to be used in each round. It also re-trains the merged model using sampled training data, thus achieving better performance with less training. Using the WISDM Smartphone and Smartwatch Activity and Biometrics Dataset [12], we examine the extreme case in which two nodes are trained using non-IID (non-independent or identical) data.

3 PROBLEM AND METHODS

The objective of online federated learning [13] is to take advantage of the additional resources and training samples that are available either in a central node or in the edge nodes.

Table 1. The distribution of instances across classes in the two examined cases (i.e. $NCTD$ and CrT).

$NCTD$		**classes**			
Subset	Number of users	**A**	**B**	**C**	**D**
T_a	20	3956	3904	3977	
T_b	20	4136	4135		4135
$Test$	10	2425	2246	2432	2429

CrT		**classes**			
Subset	Number of users	**A**	**B**	**C**	**D**
T_1	10	1799	1834		
T_a	15	3320	3237	3162	
T_b	15	2969	2970		2964
$Test$	10	2425	2246	2432	2429

In the context of this study, we implement the federated learning paradigm as depicted in Figure 1. Initially, a first model can be trained on a central node (e.g. on the cloud) and communicated to the edge nodes (step 2). The shared model is further trained on the edge nodes with the local samples that may have a completely different distribution

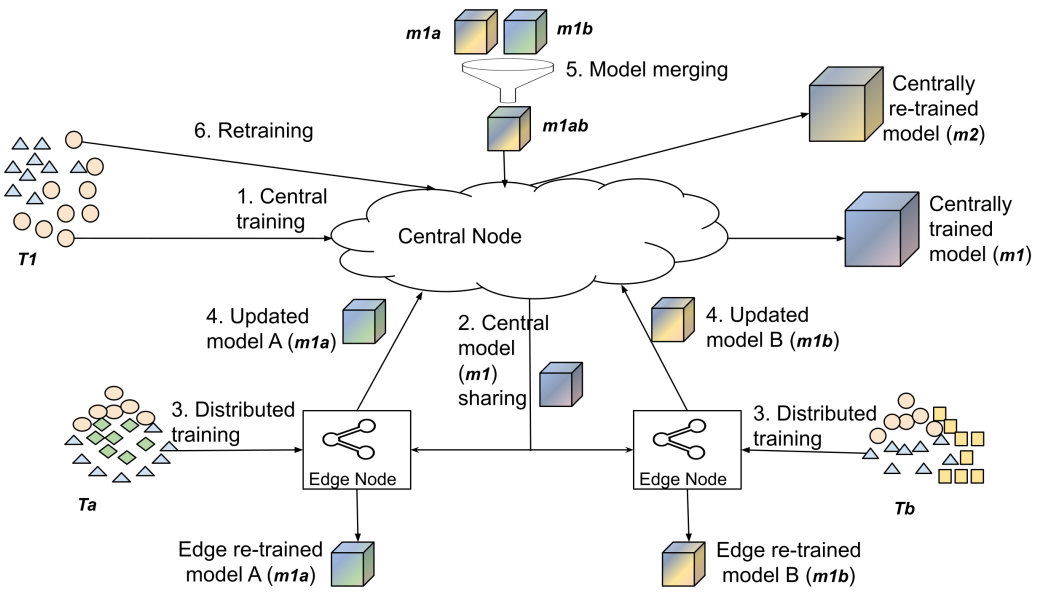


Fig. 1. The federated learning architecture examined in this work.

across classes (step 3), or even introduce previously unseen classes. The re-trained models are sent back to the central server (step 4), merged (step 5) and optionally retrained (step 6) with the centrally available training samples, before being sent back to the edge nodes for another round.

We assume a classification task with n classes (c_1, ...c_n). Training dataset T_1 is available in the central node and can be used for training or re-training at each round. Training datasets T_a, T_b, etc. are distributed on the edge nodes and used to further train the shared model (i.e. m_1) on the edge nodes. Models m_a, m_b, etc., which are trained on the edge nodes, are sent back to the central node, and a new shared model is created, by merging them. The problems that frequently arise from the class imbalance between sets T_1 and T_a, T_b, etc., is that i) either the resulting models on the edge (i.e. m_a, m_b, etc.,) deviate from the central model m_1 and introduce a bias to the merge model that leads to catastrophic forgetting of the initially seen classes, ii) or the resulting models on the edge fail to learn from new samples and thus the whole system under-performs.

In the extreme case, each edge node may introduce samples for a previously unknown class to all other nodes (including the central node. For example, in Figure 1, T_1 contains only triangles and circles, whereas T_a introduces diamonds and T_b introduces rectangles. It is important that after several training and re-training rounds the shared model (m_1), used for inference on the edge nodes is able to detect all possible classes as if (or better than when) the model was trained with all samples on the central node. In this work we examine the following alternatives:

- ***The centralised approach (C)***: The baseline approach for our case, is the centrally trained model. This is more than a simple baseline, which assumes that all the available training samples are used for training a single model in the central node.
- ***Federated learning approach (FL)***: In this alternative, the central model is initially trained using only T_1 and then is shared to the edge nodes, which use their own training samples T_a, T_b, etc. to re-train it and produce

models m_{1a}, m_{1b}, etc. respectively. The final merging of the distributed models takes place in the central node to produce a new model for sharing (m_2).

- ***Online Federated learning approach (OFL)***: It follows the principles of FL, but repeats the share-train-merge process several times (i.e. rounds). m_1 is initially trained using T_1, but in each round only a sample of T_a, T_b, etc., is used for re-training the shared model.

All the previous alternatives can slightly change if T_1 (and consequently an initial model m_1) is not available. The new model m_2 is the result of merging m_a, m_b, etc. (trained directly on T_a, T_b, etc., respectively). In the following we use ***NCTD*** to refer to this case of "*no central training dataset*". Another option is to assume that the merged model (denoted with m_{1ab} in Figure 1) is retrained using T_1 in an attempt to avoid catastrophic forgetting. We use ***CrT*** ("*central re-training*") to refer to this case.

The proposed approach builds on the OFL alternative, by adding a training data sampling technique, for the data used in each re-training step. This data sampling technique does what active learning techniques do [4], by selecting a subset of the most informative samples for training. In order to choose the most informative samples each time, we employ the model for inference first, before re-training it. Its predictions are used to filter-out the most "certain" samples and only a percentage of ambiguous samples is used for retraining. We refer to this "*Online Federated Learning with Sampling*" method as (***OFLwS***).

4 EXPERIMENTAL EVALUATION

As mentioned earlier, our dataset is based on a subset of WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set [12]. The dataset comprises the readings from smartwatch gyroscopes (on the X, Y, Z axis), for 51 test subjects that performed 18 diverse activities. The subset we use in our experiments contains observations from 50 users performing four different activities (i.e. classes): A-walking, B-Jogging, C-Stairs, D-Sitting. We selected these classes to cover a range of movements, from x-y plane only (A,B) to the z-axis too (C) and immobility (D). Instead of interpreting the input as a single time step reading, we grouped the observations in batches of 100 time steps, using a stride of 20 time steps to avoid having many overlapping batches. We then label each batch with the activity that was mostly observed inside the batch. The layers of the deep learning network comprise: a ConvLSTM 2D with ReLU activation, a Dropout at 40%, a Flatten, a Dense with ReLU, a second Dropout at 20%, and a Dense with SoftMax in the output.

In the experiments that follow, we assume i) an edge-only completely distributed setup with two nodes, trained with T_a and T_b as shown in Figure 1, and ii) a cloud-edge setup with a central node that is trained with T_1 and two edge nodes trained with T_a and T_b. The 50 users of the WISDM dataset are grouped in order to form T_1, T_a, T_b and the $Test$ dataset as depicted in Table 1 (left and right respectively for the two cases). The tables also provide the distribution of samples by class. The test set is the same in both cases.

In the absence of additional training in the central node (i.e. the $NCTD$ case) the performance of the ***centralised learning approach***, trained on the merging of T_a and T_b is ***0.62***. The recall values per class are balanced among classes: 0.66 for class A, 0.98 for B, 0.54 for C and 0.55 for D. When an additional dataset is used for training in the central node (i.e. the CrT case, which uses the merge of T_1, T_a and T_b), the accuracy of the baseline method on the $Test$ subset drops to ***0.61***. The recall values per class are: 0.84 for class A, 0.99 for B, 0.02 for C and 0.62 for D. This means that the majority of class C test instances are falsely predicted to the other three classes.

The next step is the evaluation of the ***online federated learning approach with sampling but without central training*** ***(****$OFLwS - NCTD$****)***. Starting from a randomly initialised model, which is iteratively trained in the two nodes and replaced with a merged model using Federated Averaging. Depending on the number of rounds, we split the training dataset in equally sized partitions and use one partition per round. Since the same dataset is used at all times, training in few rounds means that more data are used for training in each round and federated average is applied less frequently.

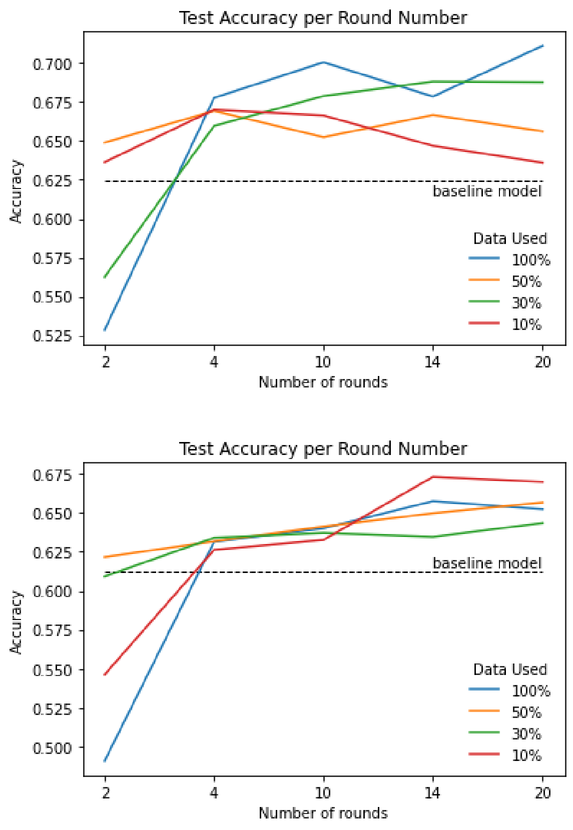


Fig. 2. The ***Accuracy*** of the Online Federated Learning with Sampling method (OFLwS) without (NCTD) and with (CrT) initial training and re-training in the central node.

In order to examine the effect of training data sampling, we run four different experiments, using 10%, 30%, 50% and 100% of the training samples available at each round. The current model is used for inference first. For each training sample we compute the entropy of probabilities to belong to each class, and keep the percentage of samples with the highest entropy (i.e. classification uncertainty) for re-training the model. The performance of the model in the test set is depicted in Figure 2 (left). The best accuracy (higher values are better) values achieved when different number of training rounds take place, are reported in the plots. Results in Figure 2 (left) show that the typical federated learning approach (without a central node) outperforms the centralised alternative. The test accuracy increases with the number of rounds, which means that a more frequent federated averaging is beneficial. Finally, using a larger percentage of training data per round leads to higher accuracy. However, this may have an impact on the total time needed for training, which is 1.5 to 6 times higher in the 100% case, than in the 10% case.

In the last experiment, we repeat the same process as before, but this time using the ***online federated learning approach with sampling and central re-training (**$OFLwS - CrT$**)***. In this case, we begin with the training of a central model m_1 and follow all the steps of Figure 1 for several rounds. The same strategy is used for selecting the most informative training samples both for re-training the distributed and the centralized (merged) model. We also use the same percentage ratio in both cases. The results are depicted in Figure 2 (right). Accuracy on test data improves when training takes place in more rounds. Once again using training data sampling is 1.5 to 2 times faster than using all data. It is interesting that using only the 10% most ambiguous samples for re-training (red line), outperforms all other methods that use more samples for training.

In order to provide a better view of the performance of the two methods (i.e. $OFLwS - NCTD$ and $OFLwS - CrT$) compared with the baseline across classes, in Figures 3 and 4 we depict the recall per class. The values for the 4 different sampling percentages, for 20 rounds of federated learning, show a drop in the recall of class C (i.e. stairs) compared to the centralised method in $OFLwS - NCTD$. This may due to the representation we employ, or the class imbalance of samples, which are finally used for training, and needs further exploration. In all other cases the proposed methods outperform the baseline.

5 CONCLUSIONS AND FUTURE WORK

This work presented a federated approach for online learning using different training sample distributions, and studied the effect of model re-training and merging frequency in the overall model performance. Although the first results are promising, there is still place for improvement in terms of using more edge nodes and experiment with more model merging techniques. This is our first attempt in implementing the online federated learning paradigm considering a cloud-edge scenario. The next plans of our work include further research on privacy preserving federated learning and on handling the issues that arise from asynchronous online learning, such as catastrophic inference and forgetting, adversarial attacks, etc.

ACKNOWLEDGMENTS

This work was supported by the project "TEACHING". TEACHING has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement N° 871385.

REFERENCES

[1] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*. PMLR, MLResearch Press, 634–643.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, et al. 2019. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046* (2019).

[3] Yanjie Dong, Georgios B Giannakis, Tianyi Chen, Julian Cheng, Md Hossain, Victor Leung, et al. 2020. Communication-Efficient Robust Federated Learning Over Heterogeneous Datasets. *arXiv preprint arXiv:2006.09992* (2020).

[4] Yuhong Guo. 2010. Active instance sampling via matrix partition. In *Advances in Neural Information Processing Systems*. Curran Associates, 802–810.

[5] William Grant Hatcher and Wei Yu. 2018. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* 6 (2018), 24411–24432.

[6] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).

[7] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 22 (2020), 2031 – 2063. Issue 3.

[8] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[9] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* (2019).

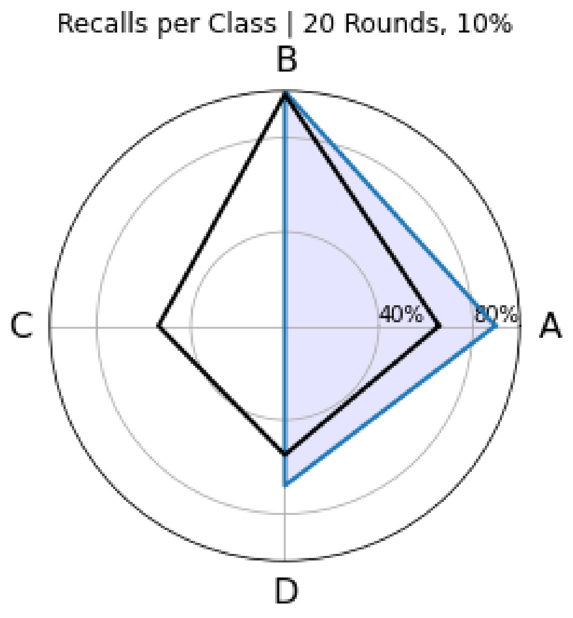
[10] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. 2019. Overcoming Forgetting in Federated Learning on Non-IID Data. *arXiv preprint arXiv:1910.07796* (2019).

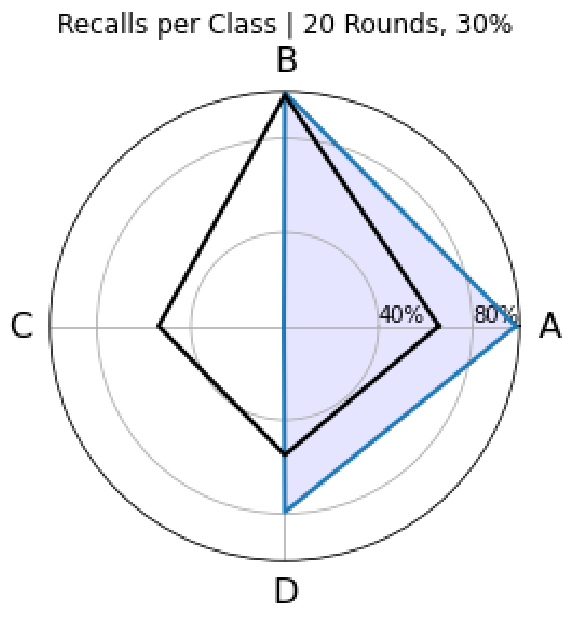
[11] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 1–11.

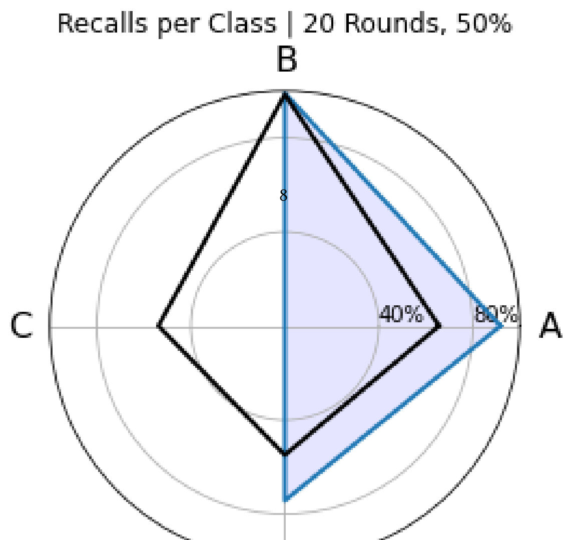
[12] Gary M Weiss. 2019. WISDM Smartphone and Smartwatch Activity and Biometrics Dataset. *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set* (2019).

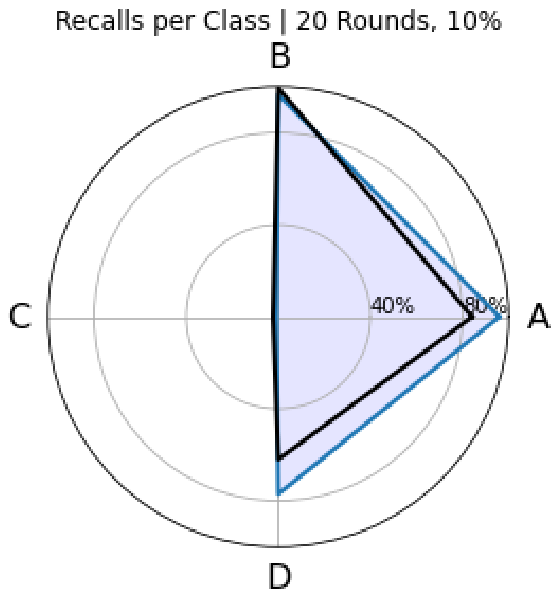
[13] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

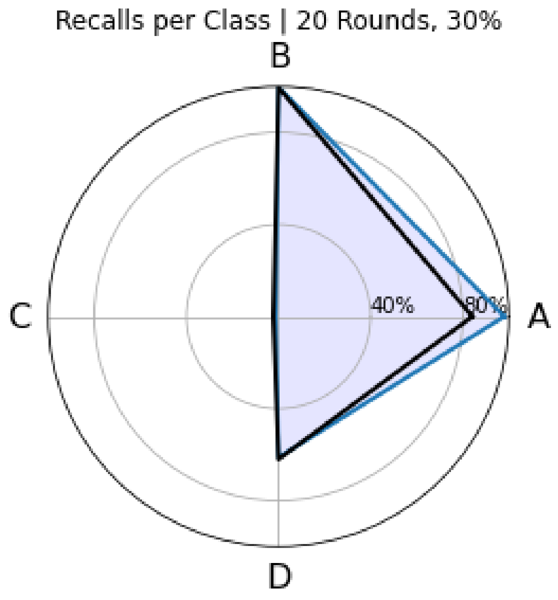
[14] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

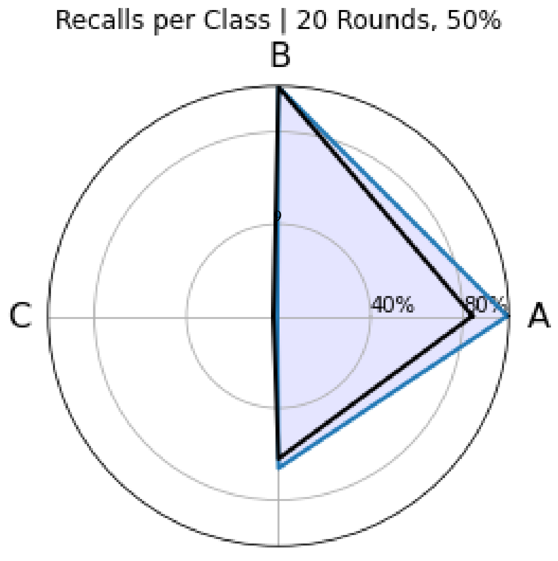

Recalls per Class | 20 Rounds, 10%
B
A
C
D
40%
80%


Recalls per Class | 20 Rounds, 30%
B
A
C
D
40%
80%


Recalls per Class | 20 Rounds, 50%
B
A
C
40%
80%


Recalls per Class | 20 Rounds, 10%
B
A
C
D
40%
80%


Recalls per Class | 20 Rounds, 30%
B
A
C
D
40%
80%


Recalls per Class | 20 Rounds, 50%
B
A
C
40%
80%