Training Effective Neural Networks on Structured Data with Federated Learning

Anastasia Pustozerova[1][0000−0003−0896−9193], Andreas Rauber[1,2][0000−0002−9272−6225], and Rudolf Mayer[1,2][0000−0003−0424−5999]

[1] SBA Research, Vienna, Austria
[2] Vienna University of Technology, Vienna, Austria

Abstract. Federated Learning decreases privacy risks when training Machine Learning (ML) models on distributed data, as it removes the need for sharing and centralizing sensitive data. However, this learning paradigm can also influence the effectiveness of the obtained prediction models. In this paper, we specifically study Neural Networks, as a powerful and popular ML model, and contrast the impact of Federated Learning on the effectiveness compared to a centralized approach – when data is aggregated at one place before processing – to assess to what extent Federated Learning is suited as a replacement. We also analyze the effect of non-independent and identically distributed (non-iid) data on effectiveness and convergence speed (efficiency) of Federated Learning. Based on this, we show in which scenarios (depending on the dataset, the number of nodes in the setting and data distribution) Federated Learning can be successfully employed.

Keywords: Federated Machine Learning, Effectiveness Evaluation

1 Introduction

The data used for training Machine Learning (ML) models often contains sensitive information, and thus must be protected from adversarial access and use. Federated Learning (FL) helps to reduce privacy risks by training ML models locally, and thus removing the need of transferring training data. It eliminates the possibility for an adversary to obtain all the training data on a centralized server. In FL, only the parameters of the learned model (e.g. weights or gradients in case of neural networks) are shared, with an aggregator or other participants.

Despite the benefits that FL entails, the effectiveness of the approach is crucial. It is imperative, for any privacy-preserving method, to achieve utility of the learned model close to a fictional, idealized (if privacy was not a concern) centralized setting. In this paper, we therefore study whether FL allows training models of a quality comparable to centralized training, and further compare it to training only on local data without collaboration. We consider different FL settings and scenarios, and formulate recommendations for using FL in different settings. We specifically focus on Neural Networks. While much of the literature focuses on image data, we take a closer look at structured (relational) data, which has many use cases e.g. in medicine, businesses and other industries.

One can distinguish FL by the coordination and aggregation strategy, into sequential and parallel learning. McMahan et al. [1] describe the *parallel* setting, where processing nodes independently and simultaneously train local models, and subsequently send them to an aggregator that computes a global model, e.g. with the *Federated Averaging* algorithm [1]. In *sequential* FL, sometimes also referred to cyclic incremental learning [2], the models are trained and shared incrementally from one node to the following in sequence. This approach does not require a dedicated aggregator, thus completely avoiding a centralized instance. Sequential learning becomes inefficient with a large number of nodes. It can, however, be a viable alternative in settings with a smaller number of nodes, e.g. when several medical institutions want to train a collaborative model, but are not able to share sensitive data.

In this work, we:

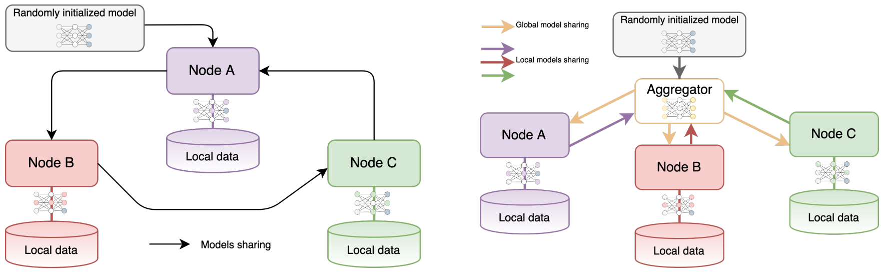
- Analyse the behaviour and performance of Federated Learning on datasets not previously considered in the literature
- Investigate how sequential and parallel Federated Learning of Neural Networks perform on structured data, and compare the effectiveness to the (idealized target) baseline results of models trained on centralized data.
- Study parallel and sequential learning with varying numbers of processing nodes in the federation, and analyze the influence on models effectiveness
- Show the impact of different distributions in the data on the model quality, and identify which Federated Learning setting (sequential or parallel) is more beneficial to use in scenarios with (i) equal distribution of data among the nodes, and (ii) non-iid data

This paper is structured as follows. Section 2 discusses related work. In Section 3, we describe the FL setup and implementation, datasets, ML models and the choice of architecture and hyper-parameters. In Section 4, we present the results of our evaluation. In Section 5, we provide conclusions and recommendations for successful federated training and provide an outlook on future work.

2 Related Work

A large share of current research in Federated Learning is dedicated to collaborative medical data processing [2, 3] due to strict privacy policies discouraging and legal regulations limiting the sharing of patients data. Federated Learning gathered significant attention as a method allowing to let data distributed on mobile devices reside there, while training effective machine learning models [4].

Federated Learning, however, poses several challenges. Communication costs can be high, especially when the number of processing nodes is large [4]. The heterogeneity of the systems (nodes) involved is a further challenge [5]. Unbalanced and not independent and identically distributed data (non-iid data) can increase the complexity of the training process, and also increase communication costs



(a) Sequential Federated Learning

(b) Parallel Federated Learning

Fig. 1. Federated Learning architectures

[6]. Moreover, security and privacy risks are an issue in Federated Learning [7]. Federated Learning allows to avoid explicit data sharing, however, the models transferred during training can leak sensitive information about local data [8].

With this wide range of challenges to address, the foremost critical remains the effectiveness of the approach – particularly whether Federated Learning results in lower quality models. Federated Learning will be considered a viable alternative only if it can achieve accuracy that is comparable, or even on par, with the ideal baseline of a centralized learning approach, and thus can be used to achieve high-quality predictions that are useful in real-world settings.

Sheller et al. [2] considered parallel and sequential learning and concluded that parallel learning gives more reliable result than cyclic incremental learning, allowing to reach 99% of the accuracy of the model trained on centralized data. In this work, we perform a more structured and broader comparison of parallel and sequential learning, using multiple datasets, and specifically testing on non-iid distributed data. Nilsson et al. [9] evaluate the effectiveness of Federated Learning on image dataset (MNIST), considering three different averaging algorithms in parallel learning. They conclude that *Federated Averaging* performs the best on non-iid data. We, therefore, use the *Federated Averaging* algorithm in the parallel learning setting in our study. McMahan et al. [1] evaluate Federated Learning performance on image and text data, and also show that *Federated Averaging* allows training effective machine learning with non-iid data. We extend this evaluation, by performing an evaluation on various structured datasets in different Federated Learning settings: sequential vs. parallel learning, iid vs. non-iid data, and a varying number of participants in the federation.

3 Study Design

In this section, we describe different Federated Learning setups, datasets and pre-processing steps, the architectures we use for neural networks and the hyperparameters choice, and finally, the settings with non-iid data.

Sequential Federated Learning (see Figure 1(a)) starts with a randomly initialized model, which is then sent to the first node in the sequence (*Node A*).

After receiving the random model, *Node A* trains it on its local data, and then sends the model to the next node in the sequence (*Node B*). The process continues until the last node in the sequence has trained the model, and the second *federated cycle* starts when the last node in the sequence sends the model to the first one. The model training process, once the federated network is set up, can be orchestrated in a peer-to-peer fashion, without the requirement of a central coordinator. This can eliminate potential single points of failure and a central point that might be subject to attacks. Also, variations on the sequence of nodes might be introduced, i.e. that it is different in each cycle.

Parallel Federated Learning (see Figure 1(b)) starts with the random initialization of a model and sending it to every node. Then, each node trains the model locally on its data and sends the trained model back to the aggregator. The aggregator averages the collected, locally trained models, and thus creates a new *global* model. We can calculate global model's weights W_{global} as an average of corresponding weights from all local models $W_k,\ k = 1, ..., n$: $W_{global} = \frac{\sum_{k=1}^{n} W_k}{n}$, where n is the number of processing nodes in the federation. After calculating a global model, the coordinator sends it back to every node for training during the subsequent *federated cycle* and so on.

Among different types of Federated Learning architecture, we focus on parallel learning as on the most spread form of Federated Learning and on sequential learning as the basic form of decentralized Federated Learning.

In our study, we first perform a grid search in centralized learning to find optimal hyper-parameters (learning rate, number of epochs, number of iterations and batch size) for the baseline comparison. We use these hyper-parameters for training models in parallel and sequential learning and, if needed, we tune some of the parameters also using grid search. We assume that each node trains their model locally with the same number of epochs in both parallel and sequential Federated Learning. We then compare the accuracy of the models trained in centralized and federated settings. We take the results of centralized training as the baseline, as it represents the ideal scenario and can serve as an upper bound of effectiveness for the final model.

We assume that each node has a fixed, unique local dataset. We follow several approaches to distribute the original (centralized) benchmark datasets utilized in our study among the nodes (see Section 3.3).

We implement our sequential and parallel Federated Learning coordination architectures using Python 3.7 and the *Pytorch* framework.

Table 1. Datasets

Dataset	# Samples	# Features	Target	Data types
Purchase-2–100	32,000	600	2,10,20,50,100 classes	binary
Location	5,280	293	30 classes	binary
Breast Cancer	683	9	2 classes	categorical
Adult	48842	14	2 classes	categorical, numerical

3.1 Datasets

To analyze a broad range of cases, we select datasets with different characteristics (cf. Table 1), to study how Federated Learning deals with these different settings. We consider eight different classification tasks (*Purchase* dataset is used with five targets) and evaluate FL performance on these tasks.

The ***Purchase*** dataset is derived from the "Acquire Valued Shoppers"[3] dataset. As in [10], we apply *k-means clustering* to create 5 different classification tasks, with 2, 10, 20, 50 and 100 classes, respectively, where each class corresponds to a group of individuals with similar purchase behavior. We denote these classification tasks as *Purchase-2*, *Purchase-10*, and so on, in the remainder of this paper. We use $30,000$ instances for training and the rest for testing).

The ***Location*** dataset is based on check-in data from the mobile phone app Foursquare, from April 2012 to September 2013[4]. Based on the description in [10], we again use *k-means* clustering to create a classification task with 30 classes, representing similar user groups. The final dataset contains $5,280$ records, representing unique users ($1,280$ are used for testing, the remainder for training). The 293 binary features represent characteristics of the places users have visited.

The ***Breast Cancer*** dataset[5], is a frequently used small-scale benchmark medical dataset. It contains 683 instances and nine categorical attributes, and a target attribute denoting the class of the instance (benign or malignant). We used 400 instances for training and 283 instances as a test set.

The ***Adult*** dataset is derived from the US Census Database[6]. The task is to predict if a person earns more or less than 50K. We use 45,000 instances for training and 3,842 for testing.

The processed and clustered versions of all the datasets are available on Zenodo[7]. We divide all dataset with five random splits into training and test data, i.e. a repeated holdout validation. In the evaluation section, all the plots and tables depict mean results among the five data splits.

3.2 Trained models

In this paper, we focus on the performance of neural networks in federated learning, and thus employ a Multi-Layer Perceptron (MLP) as a prediction model for all classification tasks. For the *Purchase* and *Location* datasets, we build on the architectures in [10], and compare our results to their baseline. The network has one hidden layer of 128 neurons, *Tanh* activation function, and a *Softmax* layer. Slightly extending the architecture from [10], we further add a *dropout* layer (0.5) for the classification tasks on the *Purchase* dataset with ten or more classes, as the regularization from the dropout layer leads to higher accuracy.

[3] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

[4] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

[5] https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)

[6] http://archive.ics.uci.edu/ml/datasets/Adult

[7] DOI: 10.5281/zenodo.4562403

As benchmark machine learning model for *Breast Cancer* dataset we follow [11]. The model we consider is a neural network with one hidden layer of nine nodes, a *ReLU* activation function on the hidden layer, and *Tanh* activation on the last layer. For *Adult* dataset, we use a fully connected neural network with one hidden layer of 64 neurons and *Tanh* activation, and *Sigmoid* activation on the last layer. We use Adam optimizer with 10^{-3} learning rate for all datasets, but *Adult*, where we use a learning rate of 10^{-4}. As a loss function for 2-targets classification tasks we use *Binary Cross Entropy Loss.* For the rest of the tasks, we use the *negative log likelihood loss* frequently applied for multi-classification tasks.

3.3 Data distribution in federation

We perform experiments with training data either (i) distributed roughly equally among the processing nodes, and also investigate the influence of (ii) non-iid data. To simulate an equal distribution, we randomly share the data among the nodes, in a way that they have the same number of instances.

To simulate a setting with non-iid data, we follow similar procedure used in [1]. We thus first sort the dataset by target label, and then split the dataset into several shards. The number of these shards is proportional to the number of nodes in the setting and different for datasets (n is the number of nodes in Federated Learning): *Breast Cancer* - $2n$ shards; *Adult*, *Purchase-2* - $4n$ shards; *Location, Purchase-10 – Purchase-100* - $10n$ shards. We simulate the size of the local training sets such that all the training data is normally distributed among the nodes. Then we assign to each node corresponding number of shards.

We compare the effectiveness of FL on non-iid data not only with the centralized baseline, which provides an upper bound of achievable accuracy, but also with the average accuracy score of the models trained just locally. This represents the alternative of not participating in FL, and provides a lower bound. We can thus investigate in which scenarios and to which extent is it beneficial to participate in federated learning, instead of just training a model on local data.

4 Effectiveness Evaluation

To evaluate the effectiveness of the models, we measure their prediction accuracy on a test set. We perform experiments on five different random splits of the data into training and test set, and report the mean and standard deviation of the accuracy on the test sets.

In the detailed analysis below, we consider two aspects of the effectiveness – the overall, final accuracy reached on centralized versus federated settings, as well as the time respectively number of iterations/cycles of training required to converge to the optimum accuracy. In some settings, this training time requirement can vary considerably between centralized and federated settings, and thus becomes an important aspect of the training parameters.

Table 2. *Location*, *Adult*, *Breast Cancer* datasets, mean and standard deviation of accuracy in % on the tests set for different number of nodes in parallel and sequential Federated Learning

	Centralized	2 nodes		8 nodes		32 nodes	
Dataset	data	Sequential	Parallel	Sequential	Parallel	Sequential	Parallel
Purchase-2	96.4 ± 0.3	96.0 ± 0.4	97.7 ± 0.2	96.4 ± 0.3	96.0 ± 0.4	96.2 ± 0.2	95.8 ± 0.4
Purchase-10	84.0 ± 0.5	83.4 ± 0.5	84.3 ± 0.6	83.7 ± 0.5	84.1 ± 0.5	83.1 ± 0.8	**80.2** ± 1.1
Purchase-20	79.0 ± 0.8	78.8 ± 0.9	79.5 ± 0.7	77.9 ± 0.7	78.8 ± 0.7	78.6 ± 1.0	**75.9** ± 1.1
Purchase-50	73.8 ± 1.0	72.2 ± 1.1	74.8 ± 1.0	72.6 ± 0.7	75.5 ± 0.7	71.6 ± 1.6	74.3 ± 1.1
Purchase-100	64.3 ± 0.6	**61.8** ± 1.8	66.6 ± 1.5	63.1 ± 1.5	66.4 ± 1.2	62.3 ± 1.0	67.3 ± 1.3
Location	80.3 ± 0.8	80.4 ± 1.1	80.6 ± 1.4	79.0 ± 0.7	81.5 ± 1.4	**76.5** ± 1.3	78.1 ± 0.7
Breast Cancer	97.5 ± 1.2	97.3 ± 1.2	95.6 ± 1.3	97.4 ± 1.1	96.8 ± 0.9	97.5 ± 1.2	96.2 ± 1.4
Adult	86.1 ± 0.7	86.1 ± 0.5	85.9 ± 0.5	86.2 ± 0.7	85.9 ± 0.4	85.6 ± 0.8	85.2 ± 0.5

4.1 Equal distribution of the data

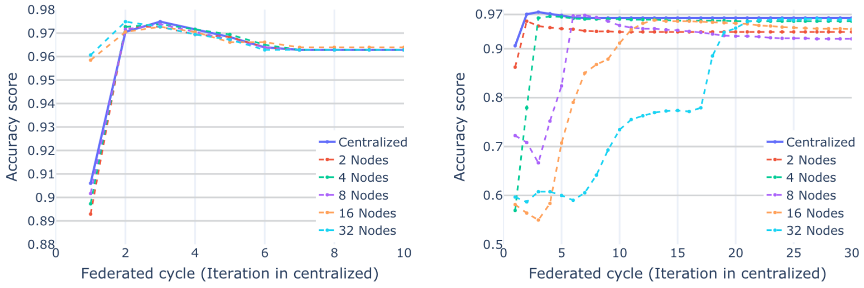
To ensure that the centralized baseline is adequate, we compare the results for *Purchase*, *Location* and *Adult* datasets in centralized setting to the benchmark from [10], for *Breast Cancer* dataset to [11]. For each dataset we train centralized models with accuracy scores close to stated in the benchmark examples.

Table 2 shows the results for the scenario of equal distribution of the data among the nodes. We observe that both parallel and sequential Federated Learning in the majority of the cases allow reaching an accuracy score close to the one achieved by the models trained on centralized data. We highlighted the few cases when the accuracy score was lower to more than 3%, than centralized baseline, e.g. sequential learning with 32 nodes on *Location* dataset. However, the deviation from the baseline accuracy is at most 4% in all considered scenarios.

Sequential learning performs worse than parallel on the classification tasks that exhibit a larger number of classes, i.e. *Location*, *Purchase-50* and *Purchase-100*. Sequential and parallel Federated Learning performed very similar on classification tasks with two targets (*Purchase-2, Adult, Breast Cancer*), both achieving an accuracy close to the baseline of centralized learning.

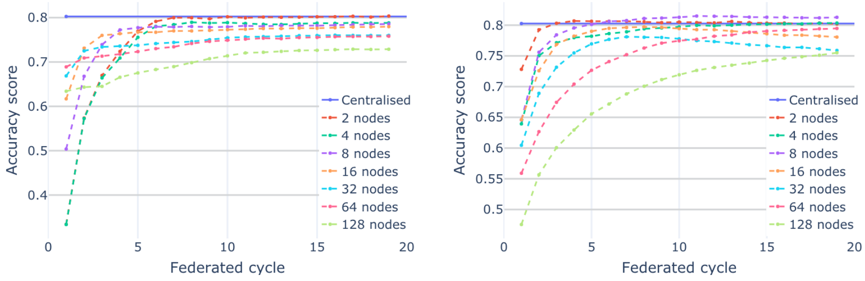
From the results on *Breast Cancer* (shown in Figures 2(a) and 2(b)) and *Adult* datasets, we notice that sequential Federated Learning manages to reach the baseline accuracy with less federated cycles than parallel. The speed of model convergence in parallel learning drops with an increasing number of nodes in the setting, e.g. with eight nodes it takes seven federated cycles to reach the baseline accuracy, while with 32 nodes, it takes 23 federated cycles (see Figure 2(b)). However, one should consider that in sequential Federated Learning a node cannot start training until the previous one finished training, and sent a model to the successor. This can make sequential federated learning considerably less efficient than the parallel, especially with a larger number of nodes in the setting.

Sequential learning performs overall worse than parallel on *Location* dataset. The quality of the final model drops up to 5% with 32 and 64 nodes, and up



(a) Sequential FL (b) Parallel FL

Fig. 2. Federated Learning on *Breast Cancer* dataset with equal distribution of the data among the nodes



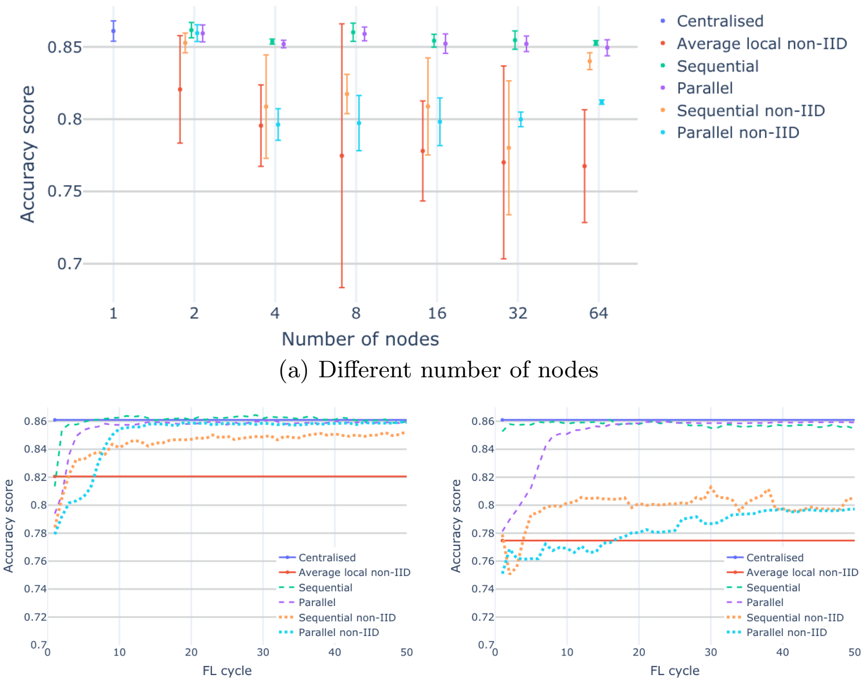
(a) Sequential FL (b) Parallel FL

Fig. 3. Federated Learning on *Location* dataset with equal distribution of the data among the nodes

to 8% with 128 nodes (see Figure 3(a)). The accuracy of the models trained in parallel learning deviates from the baseline accuracy up to 5%. This worst-case occurs with 128 nodes (see Figure 3(b)). Sequential Federated Learning is more sensitive to the increasing number of nodes than parallel.

4.2 Non-iid data

Non-iid data reduces the quality of the models trained with both parallel and sequential Federated Learning in the majority of considered settings. Moreover, it increases the number of cycles needed for models to converge. Remember that on the *Adult* dataset, sequential and parallel learning (denoted as Sequential and Parallel in the Figure 4(a)) allowed reaching similar accuracy results, only up to 1% less than centralized, with data distributed equally among the nodes. With non-iid data, both sequential and parallel reach similar scores to each other, which are, however, up to 6% worse than centralized setting. One can see that the average score of models trained only on local data is up to 5% lower (e.g. on 64 nodes) or similar (e.g. four nodes) to accuracy reached with



(a) Different number of nodes

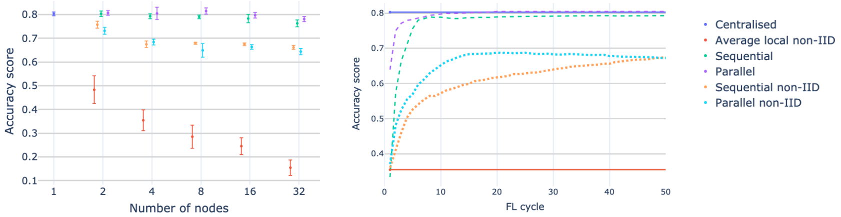
(b) FL with two nodes

(c) FL with eight nodes

Fig. 4. Comparison of the effectiveness with differing number of nodes Federated Learning on *Adult* dataset, equally distributed and non-iid data

federated learning. Further, Figures 4(b) and 4(c) shows that it takes more federated cycles for models to converge with non-iid data. In the case with eight nodes (see Figure 4(c)), models trained with sequential learning need around ten cycles to converge, and models trained with parallel learning even more than 40. Both settings only mange to reaching an accuracy 6% lower than the centralized baseline and 2% higher than lower baseline. We notice the same trend for the settings with four and more than eight nodes. In settings with two nodes (see Figure 4(b)) parallel learning performs better than sequential converging on the tenth cycle versus 20th. The accuracy of models trained with sequential learning is on average 1% less than in parallel and 3% better than the local average.

We notice that sequential Federated Learning performs better than parallel on non-iid data on the *Breast Cancer* dataset, allowing to reach model convergence for less number of federated cycles. This is especially the case with a larger number of nodes. Parallel learning also results in worse accuracy scores, for up to 5% lower accuracy than in centralized settings (sequential learning allows training models with up to 2% lower accuracy). However, both parallel and sequential learning require more federated cycles to train quality models on non-iid data (in different scenarios up to 50 more cycles).



(a) Varying number of nodes (b) FL with four nodes

Fig. 5. Federated Learning on *Location* dataset with non-iid data

Non-iid data drops federated learning performance on Location dataset even more. Figure 5(a) shows that on equally distributed data both parallel and sequential federated learning allow reaching accuracy score up to 4% lower than centralized baseline. Sequential and parallel learning had similar performance on non-iid data, which was up to 15% worse than centralized learning. We also notice that with a larger number of nodes in the setting the accuracy of the final model decreases. However, one can also see that the average of local training is dramatically decreasing with an increasing number of nodes. This can be explained by an unequal representation of each class in different nodes. In the considered non-iid setting, some of the nodes did not have instances from some classes at all and therefore could not learn any information about these classes. This is why with 32 nodes local average accuracy is only 15% while federated learning allows reaching 65-67% accuracy. Despite sequential and parallel federated learning allow reaching similar accuracy score, Figure 5(b) shows that the speed of convergence is rather different. While parallel learning converges on 15th federated cycle, sequential allows reaching similar accuracy only around the 50th cycle. Moreover, parallel learning allows training models simultaneously, which can make the overall federated learning process more efficient than sequential.

Both sequential and parallel learning perform well on *Purchase-2* dataset allowing to reach centralized accuracy with up to 1% difference. Non-iid data has less impact on models quality on *Purchase-2* dataset, in comparison to classification tasks with a larger number of classes (*Purchase-10 – Purchase-100*). While sequential learning showed better performance on the datasets with two target classes (*Breast Cancer, Adult, Purchase-2*), the parallel setting allows reaching more stable and efficient results on the classification tasks with 10 and more classes in the setting (*Location, Purchase-10 – Purchase-100*).

4.3 Main findings and recommendations

From our experimental evaluation on eight different classification tasks on structured, relational datasets, we can draw the following findings:

- In the settings *with equal distribution of the data among the nodes, Federated Learning allows reaching accuracy close to centralized learning* (at most up to 5% difference in accuracy), with similar hyper-parameters and the same model architecture used in centralized settings.

- *Non-iid data demands more federated cycles for a model to converge than equally distributed data.* The speed of convergence decreases with an increasing number of nodes in federated learning. It is more difficult to reach baseline accuracy with non-iid data for both sequential and parallel federated learning, however both give *a higher score comparing to only local training.*
- We note that *classification tasks with two classes are less influenced by non-iid data*, than tasks with a larger number of target classes. The effect is especially pronounced in the settings with 16 nodes and more on all considered datasets. This is likely caused by the fact that with many classes and many nodes, the number of data items per node in each class is becoming very small, and it is thus very difficult for the individual models to train a generalizing model on that class.
- Parallel Federated Learning performs better than sequential on classification tasks with a large number of classes – especially with a larger number of nodes in the setting (more than four nodes). It allows to reach higher accuracy of the models, and takes less federated cycles for models to converge. That holds for both equally distributed and non-iid data. It is thus a *recommendation to use parallel Federated Learning on classification tasks with ten and more classes*
- Sequential learning showed good performance on classification tasks with two target classes, allowing to reach close to baseline accuracy (at most up to 2% lower), with less than ten cycles even with non-iid data. This approach, therefore, is a viable solution allowing to avoid centralized aggregation of the models as in parallel learning. It is thus a *recommendation to use sequential Federated Learning on two-classes classification tasks.*

5 Conclusion

Federated Learning allows to perform privacy-preserving machine learning on sensitive data, and thus offers an alternative to settings where data needs to be centralized and/or anonymised for processing. Federated Learning needs to achieve the effectiveness similar to centralized setting to be considered a viable alternative. In this paper, we thus studied the effectiveness of classification algorithms on multiple datasets.

We showed that Federated Learning allows reaching a baseline accuracy in settings with equally distributed data, comparable to models trained on centralized data (at most with up to 5% drop in accuracy score). The hyper-parameters applied in centralized learning (e.g. the number of epochs, iterations, learning rate) can be used to train effective models in sequential and parallel Federated Learning. With non-iid data, training good quality models with Federated Learning can results in significantly lower accuracy and can entail higher communication costs due to the larger number of federated cycles needed for the model to convergence.

Future work will focus on extending this analysis to additional datasets, different machine learning algorithms and a thorough investigation of privacy threats in Federated Learning.

6 Acknowledgments

This work was partially funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078 (FeatureCloud). SBA Research (SBA-K1) is a COMET Centre within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the federal state of Vienna. The COMET Programme is managed by FFG.

References

1. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y. Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2017. PMLR.
2. Micah J. Sheller, G. Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. *International Workshop on Brain Lesion (BrainLes), in conjunction with MICCAI*, 2018.
3. Nicola Rieke, Jonny Hancox, Wenqi Li, et al. The future of digital health with federated learning. *npj Digital Medicine*, 3(1), Sep 2020.
4. Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.
5. Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. *IEEE International Conference on Communications (ICC)*, 2019.
6. Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.
7. Lingjuan Lyu, Han Yu, Jun Zhao, and Qiang Yang. *Threats to Federated Learning.* Springer International Publishing, Cham, 2020.
8. Stacey Truex, Ling Liu, Mehmet Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.
9. Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Workshop on Distributed Infrastructures for Deep Learning.* ACM, 2018.
10. R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017.
11. Alexis Marcano-Cedeño, Fulgencio S. Buendía-Buendía, and Diego Andina. Breast cancer classification applying artificial metaplasticity. In *Bioinspired Applications in Artificial and Natural Computation*, Berlin, Heidelberg, 2009. Springer.