Table of contents :

Table of Figures and Table :

Problem statement

Federated learning can be used to solve problems related to medical diagnosis since the data is quite private and the federated learning can allow us to use data from different hospitals without sharing data at a single place.We would be performing the federated learning with differential privacy on two different types of datasets ,first one being a text based small data set and second one being medical imaging dataset. We would be analysing how well they are performing as we increase or decrease the number of clients in the system.

Literature Review of Existing Methods

2018 **intel , university of pennsylvania** used publicly available brain tumour segmentation (BraTS) dataset.MRI brain scans from patients with gliomas were taken.Each of the abnormal findings was manually annotated by radiologists using three distinct labels U-Net architecture at server client federated learning for training and validations. Data split and tested in 2 ways: First split : randomly selected data and allocated to each client Second split: assigned data to institutions from which they were initially selected During implementation multiple clients received the current version of the model from the central server and updated the model using the federated aggregation.

Found that the performance scores same as the complete dataset model which would be trained completely in one single system.

Nvidia , king's college London :Training performed on the clara Train SDK.Using BraTS 2018 dataset, differential privacy preserving learning. In above examples the problem of not having enough data was bypassed. **[1]**

In the **research paper [2]** authors focused on differential privacy related concepts and encrypted computation.Three main concepts discussed in the paper were:

1) **Anonymization and Pseudonymization** : Remove personal data from images and replace it with artificial data .Adds to complexity, security of the search tables also taken care
2) **Differential privacy** : Avoid statistical analysis.Like a person cannot say that weather this particular entity was used to derive conclusion
3) **Homomorphic encryption**: The algorithm can be used to encrypt and decrypt medical images until the benefits of homomorphic encryption in providing effective protection to original data are understood. When a model has a sole owner, homomorphic encryption allows the owner to encrypt their model such that untrustworthy third parties cannot train or use it without stealing it.

In **research paper [3]** they focused on the data heterogeneity concepts for federated learning.Main concepts were as following:

1) Thorough research to study the impact of a taxonomy of data heterogeneity regimes on several widely used federated learning methods with medical image data.
2) Showed that the performance of the federated learning methods in our study degrades with the increasing degrees of data heterogeneity, and the rate of decrease in performance is determined by the degree of deviation from homogenous distributions.
3) Proposed a variety of optimization strategies to mitigate the performance loss for quantity skew and label distribution skew, including weighted average strategy for data quantity skew, and weighted loss strategy for label distribution skew.
4) the influence of the Batch Normalization (BN) for FedAVG, that averaging the mean and variance of BN across institutions during FedAVG training is a simple and flexible alternative to mitigate skew-induced performance loss of BN

DATASET : ADNI ,Pet scans

Retina 17,563 pairs of right and left color digital retinal fungus images

Quality skew

-> Performance of the FedSGD and CWT drop with increase in heterogeneity

->FedSGD+WL : gradients from each institutions not treated equally but proportional to the institutional sample size, improved performance was observed

Label distributions skew:

Mainly because of diff annotations ,but also exists otherwise when one disease can be inherently more common in certain set of people

All algos show reduction in performance

Using the class weighted cross entropy loss sees reduction in degradation , but in some splits performance degrades even more

->Modified batch norm where we updated the mean variance across institutions in the training

FedAVG+WL+BN superior performance than FedAVG+WL

Image distribution skew:

No standardization on how to images are collected

super-resolution, image denoising and histogram matching may be applied to deal with the imaging acquisition skew, which is not included in paper

Another major and interesting work was **HDAFL (Heterogeneous Data-Aware Federated Learning) [4]:**

New technique where the generic parameters are updated in the global manner and specific are updated only locally .Gave better performance in non iid data.

A work done recently,**FEDBN [5]** showed that how we can get better results on IID data if local batch normalization is used. Authors used medical imaging to compare the fedbn, fedavg and fed prop.They updates the non-BN layers using FedAvg, without modifying any optimization or aggregation scheme. This approach has zero parameters to tune, requires minimal additional computational resources, and can be easily applied to arbitrary neural network architectures with BN layers

Important work that worked the concept of differentially private learning using homomorphic encryption was **[6] Dopamine** Authors performed a customization of DPSGD (Differentially private stochastic gradient descent) for FL, which, in combination with secure aggregation by homomorphic encryption, can establish a better privacy-utility trade-off than the existing approaches.

Problems Identified

Expensive communication: There would be bottleneck bandwidths so because of this communication the model training would be faster in the local systems rather than multiple systems.

System heterogeneity:The devices in the system would be having different nature , storage wise , energy consumption etc. The dropping of devices would also be there.

Statistical heterogeneity: Different devices with different numbers of data points violate independence and are identically distributed. This have been worked upon in some studies suggested above, but some improvements can still be made.

Sharing the updates of parameters like gradient values may **reveal sensitive info** about the particular data units.

We would be implementing the **differentially private** machine learning on the text as well as medical imaging dataset .Using this we would be further analysing how well the system is able to expand.That is if we would increase the number of clients on the system then how would the accuracy would be affected.

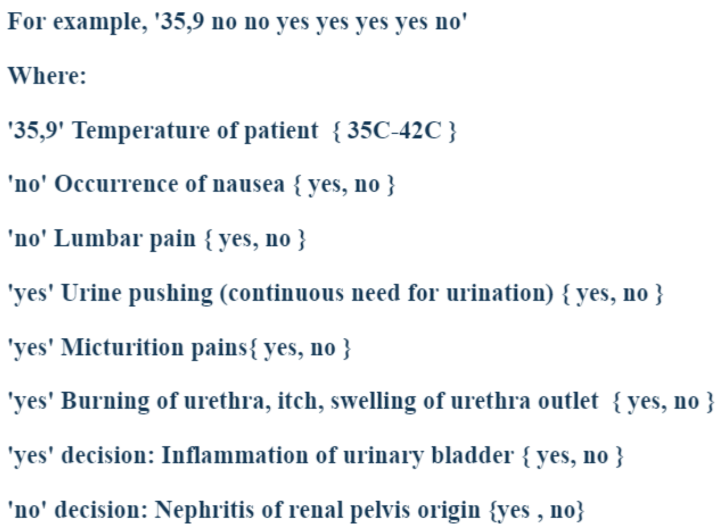
Datasets

Dataset 1 :

Acute Inflammations Data Set : Dataset was from :

https://archive.ics.uci.edu/ml/datasets/Acute+Inflammation

The dataset consist of information of 2 diseases Inflammation of urinary bladder and nephritis of renal pelvis origin.The dataset consists of features and 2 columns that would be telling whether the disease was detected or not in this case Data set is multivariate , having 120 instances , number of attributes : 6.The data is in an ASCII file. Attributes are separated by TAB.Each line of the data file starts with a digit which tells the temperature of the patient.



For example, '35,9 no no yes yes yes yes no'

Where:

'35,9' Temperature of patient { 35C-42C }

'no' Occurrence of nausea { yes, no }

'no' Lumbar pain { yes, no }

'yes' Urine pushing (continuous need for urination) { yes, no }

'yes' Micturition pains{ yes, no }

'yes' Burning of urethra, itch, swelling of urethra outlet { yes, no }

'yes' decision: Inflammation of urinary bladder { yes, no }

'no' decision: Nephritis of renal pelvis origin {yes , no}

Fig 1: Dataset 1 structure

Dataset 2:

This was a medical imaging dataset.The dataset contains the images of the chests and we have to determine whether the patient has pneumonia or not .The dataset has 5856 files which are divided into 3 folders train, test and validation set , where each set has pneumonia and normal images.

Solution implemented

PART1 : Working on the text data :

Federated learning system is made to diagnose the two acute inflammations of bladder. Medical dataset contains data of 2 diseases :Inflammation of urinary bladder and nephritis of renal pelvis origin. The dataset strictly required privacy though we were able to achieve good results without it, the data was linearly separable but the issue is that the data is very personal to the patients so to maintain this property we use federated learning

Pysft and OpenMind are used for this purpose .The privacy is protected using the federated learning and the robust ML model is also made.Pysft has tools that allow us to use the federated learning openMind ensures the differential privacy and also that the communications between the clients and the aggregator.

Features : Temperature of patient, Occurrence of nausea, Lumbar pain, Urine pushing (continuous need for urination), Micturition pains, Burning of urethra, itch, swelling of urethra outlet.

Decision: Inflammation of urinary bladder, Nephritis of renal pelvis origin
Firstly we used a **logistic regression** model

```
input_size = 6
learning_rate = 0.01
num_iterations = 20000
```

Firstly we trained a model to diagnose **Inflammation of Urinary Bladder.** As in fig 1). we can see that the training loss finally is zero and accuracy is 100 %.
Similarly for the **Nephritis of Renal Pelvis Origin** we got 100 percent accuracy fig 2).

Federated learning model

We assumed that there are n hospitals. (The dataset will be split in n parts , randomly). The n hospitals cannot share the cases of their patients because they are competitors and it is necessary to protect the privacy of patients.

So we use federated learning technique.Overall we used 500 iterations. In one iteration we would make a copy of the shared model and send it to all the n hospitals.In each hospital / model in local site training would be done using the local dataset , a total of 5 iterations would be done locally.Each model would perform some improvement in its own direction.
Once this is done the local losses and accuracies are computed to keep track of them.Locally prepared models would then be sent to a trusted aggregator that will perform average of all updates , this is now the updated model that will be sent to each local site at start of each iteration.
So only ML model is shared and patient data at each hospital is kept private and each model is performing updates in the local manner.More data is available and also data is not shared

We are constructing two learning curves: **Training Losses versus Iterations** and **Training Accuracies versus Iterations**.Finally, we compute the testing accuracy of the final model with the testing dataset we separated from the beginning.

We would be using applying the same technique on the 2,4,6 number of hospitals and results would be compared.

PART2 : Working on the image data:

1) Preparing the dataset : Data set is already strutructed into the train,test and valid set.The data was loaded as it is into

2) Image Augmentation : Performed to increase size of the dataset , we use OneOf method given by pytorch and use techniques like : HorizontalFlip,IAAAdditiveGaussianNoise() ,MotionBlur and a number of other techniques.We group them into functions and associate a probability as OneOf function parameter.

3) Visualizing train images : Random 10 images are visualized to check if the dataset is being linked properly or not ,using the plt() function.

4) Model used : Convolutional neural network is used.Model is stated below :

```
 Net(

(conv1): Conv2d(3, 20, kernel_size=(5, 5), stride=(1, 1))

(conv2): Conv2d(20, 50, kernel_size=(5, 5), stride=(1, 1))

(fc1): Linear(in_features=140450, out_features=500, bias=True)

(fc2): Linear(in_features=500, out_features=10, bias=True)

     )
```

First at the start we have a conv2d layer with number of channels in the input image = 3 , number of channels produced by the convolution = 20,kernel size = 5x5 and stride of 1.

The second conv2d layer has input = 20 output = 50 and kernel size and stride the same as before.

Followed by it is the linear connected layer with in_features and out_features 140450 and 500 respectively.Last fully connected layer has in_features as 500 and out_features as 10.

5) Preparing training and validation PyTorch loaders: Using the function Dataloader() in pytorch we would then make train and test loaders using the batch size = data_size where data_size is the data available with each client which is math.ceil(len(dataset) / len(workers))

6)Training: Federated learning using the syft and federated loaders is done by making **batch size** of **128** and then distribute over the given n clients.**CrossEntropyLoss()** is used as the minimizing criteria for the loss function with **learning rate = 0.0005** and optimizer **stochastic gradient descent**.Training is done for **10 epochs** and testing accuracy is calculated .

We performed this experiment by changing the number of clients from 2 to 6 and results are shown in the result section below.

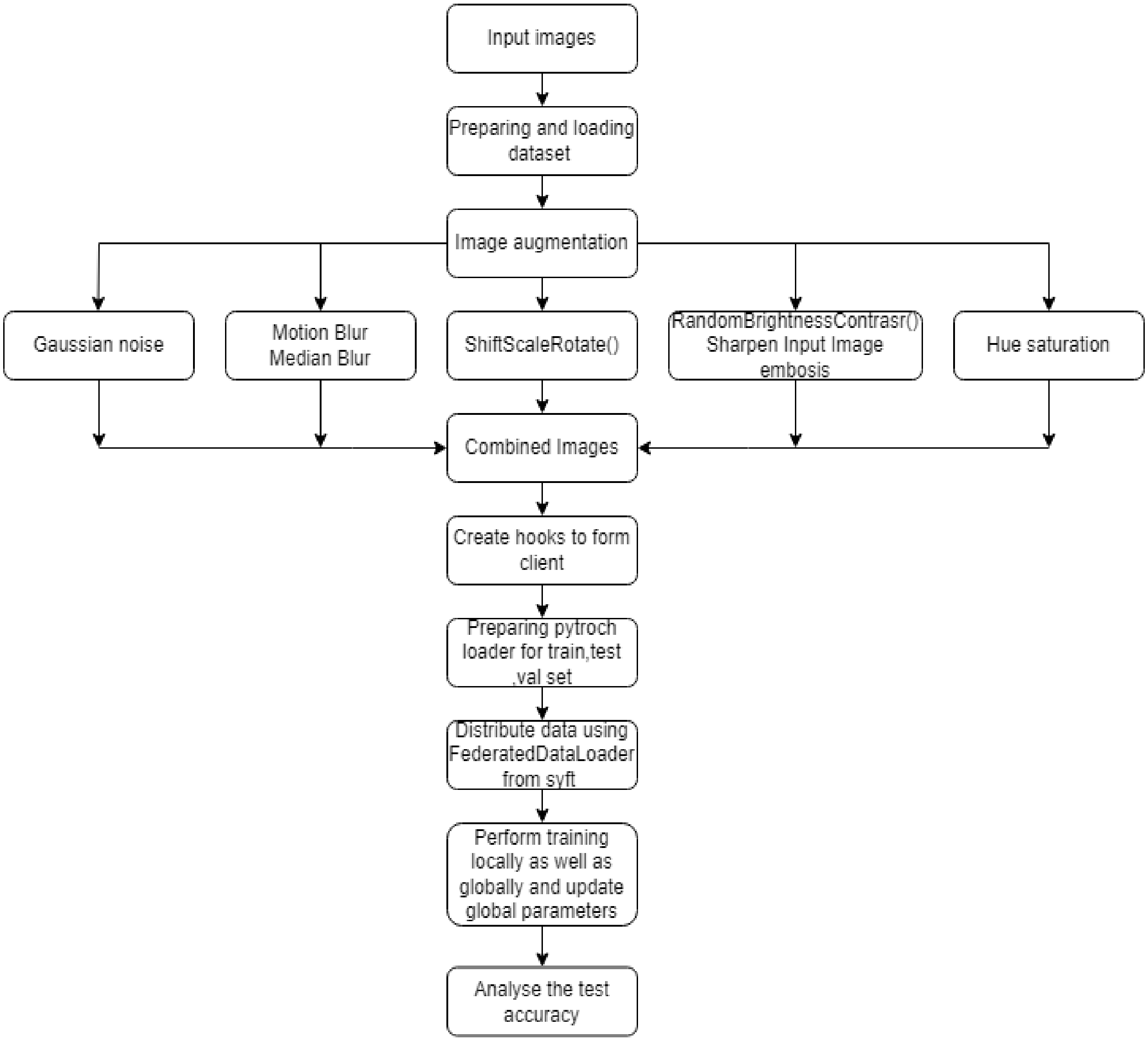

Input images
Preparing and loading dataset
Image augmentation
Gaussian noise
Motion Blur
Median Blur
ShiftScaleRotate()
RandomBrightnessContrasr()
Sharpen Input Image
embosis
Hue saturation
Combined Images
Create hooks to form client
Preparing pytroch loader for train,test ,val set
Distribute data using FederatedDataLoader from syft
Perform training locally as well as globally and update global parameters
Analyse the test accuracy

Fig 2: Complete implementation pipeline

Outcomes:

On dataset 1:

The Training accuracy on both the sets have been shown below when basic implementation is done.The graphs show that accuracy vs iterations and clearly state that we are able to get a high accuracy of 100 percent.

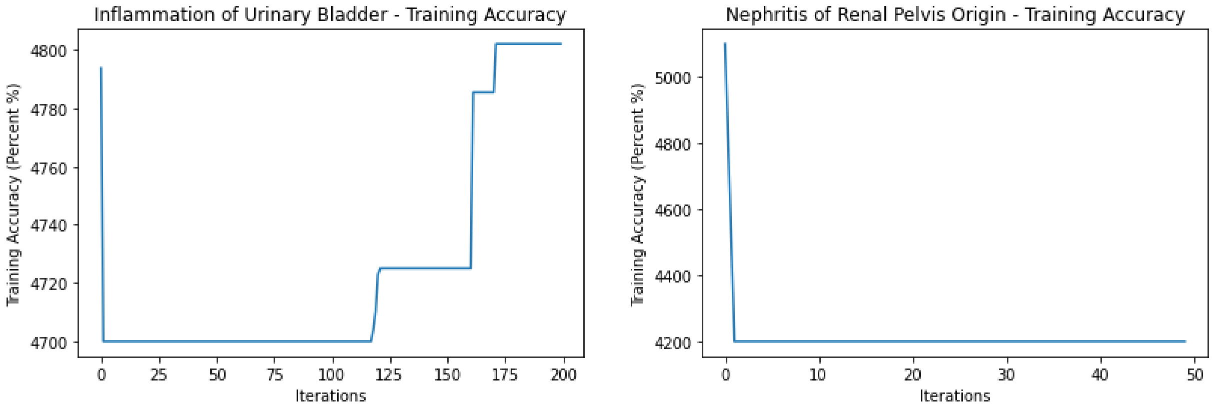


Fig 3: Training accuracy in both the classification task without federated learning

TASK	2 Clients	4 Clients	6 Clients
Inflammation of Urinary Bladder	45.83 %	58.53 %	54.17 %
Nephritis of Renal Pelvis Origin	66.67 %	66.67 %	66.67 %

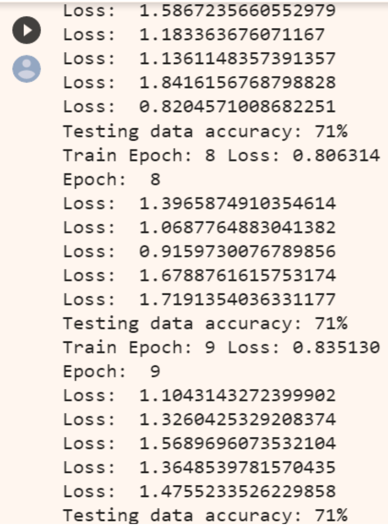
Table 1: Comparison of accuracies for different number of clients

We used only 200 iterations for the comparison purpose though when we used a suitable number of iterations that is 1000 used we were able to get **100 percent test accuracy** for all the clients:

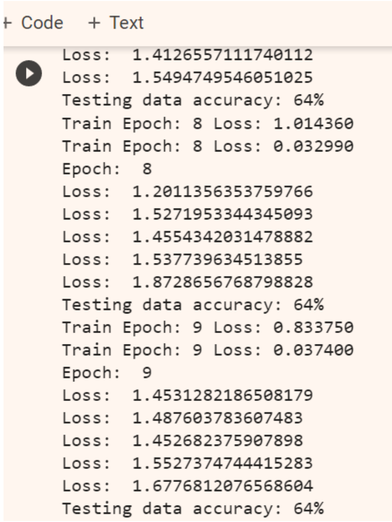
On dataset 2:

We ran system for 10 epochs with 2,4,6 clients and results reported as shown here.

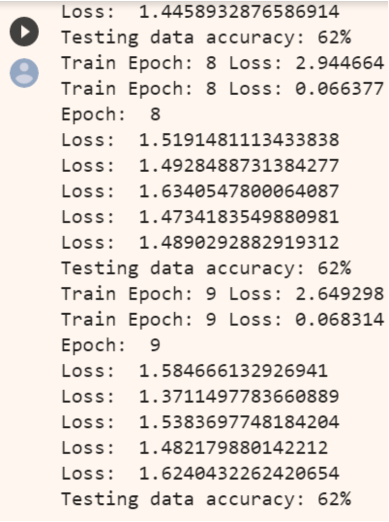
Clients 2	Clients 4	Clients 6
71%	64%	62%



```
Loss:  1.5867235660552979
Loss:  1.183363676071167
Loss:  1.1361148357391357
Loss:  1.8416156768798828
Loss:  0.8204571008682251
Testing data accuracy: 71%
Train Epoch: 8 Loss: 0.806314
Epoch:  8
Loss:  1.3965874910354614
Loss:  1.0687764883041382
Loss:  0.9159730076789856
Loss:  1.6788761615753174
Loss:  1.7191354036331177
Testing data accuracy: 71%
Train Epoch: 9 Loss: 0.835130
Epoch:  9
Loss:  1.1043143272399902
Loss:  1.3260425329208374
Loss:  1.5689696073532104
Loss:  1.3648539781570435
Loss:  1.4755233526229858
Testing data accuracy: 71%
```



```
Loss:  1.4126557111740112
Loss:  1.5494749546051025
Testing data accuracy: 64%
Train Epoch: 8 Loss: 1.014360
Train Epoch: 8 Loss: 0.032990
Epoch:  8
Loss:  1.2011356353759766
Loss:  1.5271953344345093
Loss:  1.4554342031478882
Loss:  1.537739634513855
Loss:  1.8728656768798828
Testing data accuracy: 64%
Train Epoch: 9 Loss: 0.833750
Train Epoch: 9 Loss: 0.037400
Epoch:  9
Loss:  1.4531282186508179
Loss:  1.487603783607483
Loss:  1.452682375907898
Loss:  1.5527374744415283
Loss:  1.6776812076568604
Testing data accuracy: 64%
```



```
Loss:  1.4458932876586914
Testing data accuracy: 62%
Train Epoch: 8 Loss: 2.944664
Train Epoch: 8 Loss: 0.066377
Epoch:  8
Loss:  1.5191481113433838
Loss:  1.4928488731384277
Loss:  1.6340547800064087
Loss:  1.4734183549880981
Loss:  1.4890292882919312
Testing data accuracy: 62%
Train Epoch: 9 Loss: 2.649298
Train Epoch: 9 Loss: 0.068314
Epoch:  9
Loss:  1.584666132926941
Loss:  1.3711497783660889
Loss:  1.5383697748184204
Loss:  1.482179880142212
Loss:  1.6240432262420654
Testing data accuracy: 62%
```

Table 2: Comparison of accuracies for different number of clients in X-ray data

Results

We can clearly observe that in the dataset 1 there is a little irregular increase and decrease of the accuracies in first classification but it is constant in the case of second classification task, the change is also very less so we can in a way that it is able to scale well in case of increase of clients.

While in the dataset 2 there is a regular decrease, One of the reason could be that the feature extraction in case of the image data is a part of the cnn only so in a way there are chances that the features are not getting extracted in a way that could be done if we had whole data .

Though the difference is not that much so that we can say that system is not scalable

Conclusion and future work

We were able to implement a good and highly accurate system for both the datasets.We were able to show that the systems are scalable and reduction in accuracy is not that much with increase in clients.

In future we can try to improve scalability more for the imagining data , where feature extraction could be an important point of concern to get better results.

References

[1] Dianwen Ng 1, et al. “Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets.” *National Center of Biotechnology Information*.

[2]Unnati Shah, et al. “Maintaining Privacy in Medical Imaging with Federated Learning, Deep Learning, Differential Privacy, and Encrypted Computation.” *6th International Conference for Convergence in Technology*.

[3] Daniel L Rubin. “An Experimental Study of Data Heterogeneity in Federated Learning Methods for Medical Imaging.”

[4] HDAFL (Heterogeneous Data-Aware Federated Learning)

[5] FEDBN: Federated Learning on Non-IID Features Via Local Batch Normalization.

[6] Generative Models for Effective ML on Private, Decentralized Datasets

[7] FedML: A Research Library and Benchmark for Federated Machine Learning

[8] Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge.

[9] Central Server Free Federated Learning over Single-sided Trust Social Networks

[10] Mohammad Malekzadeh. “Dopamine: Differentially Private Federated Learning on Medical Data.”