Journal of Medical Systems (2023) 47:99
https://doi.org/10.1007/s10916-023-01994-5

ORIGINAL PAPER

A Federated Learning Approach to Tumor Detection in Colon Histology Images

Gozde N. Gunesli[1] · Mohsin Bilal[1] · Shan E Ahmed Raza[1] · Nasir M. Rajpoot[1]

Received: 6 June 2023 / Accepted: 7 September 2023 / Published online: 16 September 2023

Abstract
Federated learning (FL), a relatively new area of research in medical image analysis, enables collaborative learning of a federated deep learning model without sharing the data of participating clients. In this paper, we propose *FedDropoutAvg*, a new federated learning approach for detection of tumor in images of colon tissue slides. The proposed method leverages the power of dropout, a commonly employed scheme to avoid overfitting in neural networks, in both client selection and federated averaging processes. We examine *FedDropoutAvg* against other FL benchmark algorithms for two different image classification tasks using a publicly available multi-site histopathology image dataset. We train and test the proposed model on a large dataset consisting of **1.2 million image tiles** from **21 different sites**. For testing the generalization of all models, we select held-out test sets from sites that were not used during training. We show that the proposed approach outperforms other FL methods and reduces the performance gap (to less than 3% in terms of AUC on independent test sites) between FL and a central deep learning model that requires all data to be shared for centralized training, demonstrating the potential of the proposed *FedDropoutAvg* model to be more generalizable than other state-of-the-art federated models. To the best of our knowledge, ours is the first study to effectively utilize the dropout strategy in a federated setting for tumor detection in histology images.

Keywords Tumor segmentation · Histology images · Federated learning · Neural model aggregation

Introduction

In recent years, deep learning methods have shown excellent predictive performances in many different tasks including those in computational pathology [1]. A major drawback to these approaches is the need for large amounts of data to train the networks. This drawback is even more obstructive in the medical field, as medical data are difficult to access and their sharing may be subject to legal and ethical limitations.

Federated learning (FL) [2, 3] allows to overcome these challenges. While in traditional deep learning approaches, all data is required to be co-located in a single centralized pool, during the collaborative training of a model by FL, each participating site (also referred to as client, or collaborator) train the model locally on their own servers and only share their model parameters with an aggregation server. By enabling collaborative training of the deep learning models without exchanging the datasets, FL offers a solution to data ownership and governance issues [4]. Existing FL methods comprise several rounds of local training and aggregation steps. In each round of the federated training process, each data holder trains a model for some number of epochs on their local dataset. The local data holders then send their trained models to the central aggregation server for model aggregation. The aggregated model i.e., the consensus model, is sent back to the data holders for further training rounds.

Model aggregation is an important step in the training of a federated model. The most commonly used method of model aggregation in existing FL studies is Federated Averaging (FedAvg) [3]. FedAvg performs a weighted averaging of local

✉ Gozde N. Gunesli
Gozde.Gunesli@warwick.ac.uk

✉ Nasir M. Rajpoot
N.M.Rajpoot@warwick.ac.uk

Mohsin Bilal
Mohsin.Bilal@warwick.ac.uk

Shan E Ahmed Raza
Shan.Raza@warwick.ac.uk

[1] The Tissue Image Analytics Centre, Department of Computer Science, University of Warwick, Coventry, UK

model parameters (the gradients) to obtain a global consensus model at each round. The weights in this case are determined based on the number of training samples of each local data holder. Li et al. [5] argued that local models are often substantially different from global models because of the heterogeneous and imbalanced nature of the datasets. Therefore, they proposed FedProx, which contained a proximal term in the loss function to restrict the effects of local training and prevent divergence from the global model parameters.

For the aggregation step, weighting clients' models according to the number of training samples, as in FedAvg and FedProx, is reasonable since local training of a client with a bigger training dataset size has a higher chance of being more useful, as it is exposed to more data. However, with this approach, the aggregated model might be overfitting to the data distribution of the clients which have the most training samples, while other clients are always decidedly contributing less even though learning their distributions might be better in terms of statistical variability. The approach of FedProx might even increase this negative effect by adding a regularization term to penalize the model parameters changing too much from the global model during local training rounds.

Personalized FL (PFL) is another approach [6–9] which is proposed to help trained models perform better on the local datasets. In these approaches, each client has their own version of the model during training and they keep some specific layers private. These private (also called 'personal') layers are not updated using the global consensus model parameters. In [7], it is proposed to keep the parameters of the batch normalization layers private. In [6, 8], they have proposed to train some layers (or blocks) of the neural networks privately, while the parameters of the non-private layers are getting updated at each round using the global model parameters sent by the server. Although these PFL approaches may help the models learn the clients' local training data distribution better, these models may perform poorly on unseen data.

Some approaches [10], involve sharing some partial information about the data to mitigate the effects of the heterogeneity of the datasets, compromising privacy to some degree. Since providing data privacy is the main concern of FL, sharing data might not be preferable and/or feasible for many applications and this type of approach is not considered in this study.

Why dropout in FL? Local medical image datasets can be heterogeneous and unbalanced in terms of number of samples. Besides, the diversity of samples collected by different institutions could differ by a large margin. In such scenarios, the approach of FedAvg and FedProx, weighting the contributions of each local model by their data size, may have significant limitations. Since it may not be known beforehand which private local dataset(s) may generalize better for the test set of another site, measuring the contributions of individual sites and accurately weighing them may not be feasible. Dropout mechanisms, their approximation properties, and their effectiveness by introducing noise to the network training, by sampling the data or training an ensemble of models have been extensively studied in the literature [11–14]. Similarly, the use of dropout can be seen as an approximation method in FL training, in which approximation errors will cancel each other. As such, instead of determining model contribution weights beforehand based on data size or having personal models which will not perform well on unseen data, using dropout mechanisms would introduce additional randomness to client selection and parameter aggregation. Therefore, we propose introducing dropout strategies for global model aggregation and client participation in FL training to mitigate the complexities of learning from imbalanced and heterogeneous datasets from various clients.

The proposed FedDropoutAvg is different from the well-known dropout technique by Srivastava et al. [14]. The conventional dropout [14] is used for a few network layers during training of the local model to reduce overfitting on the local dataset, having no effect on the FL aggregation step. On the other hand, the proposed approach consists of two main components that both occur at the server: randomly dropping parameters for aggregation and dropping clients. Dropping parameters happens at the end of each round, where some random parameters from each model are not included in the aggregation. Client dropout happens at the beginning of each round, where the server randomly selects clients to participate and only sends the global model to them, causing dropped-out clients to not participate in that round's aggregation.

In this paper, we propose Federated Dropout Averaging (FedDropoutAvg), a new FL aggregation method with the objective of obtaining a global federal model that improves the model performance by adjusting dropping out of the parameters of locally trained models before aggregation and also randomly dropping out some clients at each round. Our approach is inspired by FL model training with sensitive user data on mobile devices [2, 3], where thousands of clients participate and several clients may also get dropped out at each training round due to various reasons and constraints like unstable connectivity or efficiency [3, 5, 15–17]. However, to the best of our knowledge, the impact of client dropout in FL training for medical image analysis tasks remained unexplored.

We explore the comparative performance of different FL model aggregation strategies on multi-institutional histopathology image datasets using federated models in a simulated decentralized setting for real-world data, using not only test data from sites selected for training but also the held-out test data from completely different (independent) sites outside the federation. We present a comparative evaluation of the proposed method with locally trained models, the models trained in a centralized manner, two major FL model aggregation methods and various personalized FL approaches for two different classification tasks on these datasets.

Specifically, our work makes the following novel contributions:

- We present *FedDropoutAvg*, a novel FL approach for federated training of deep learning models for histopathological image classification. This is the first study to propose and explore:
  - random dropout of sites (clients) for each round of federated training,
  - and random dropout of parameters of locally trained models for aggregation into a federated model for medical image analysis.
- We demonstrate the effectiveness of our method using a large multi-site dataset. The first dataset used in this study consists of 1.2 million images from 21 different sites. The individual datasets of these sites are imbalanced in terms of the number of samples, patients, and the number of positive and negative images and contain significant variation in the image data (color, brightness, focus) – the so-called *domain shift*, as can be observed in Fig. 2 – due to variations in scanning and staining parameters.
- We show that FedDropoutAvg outperforms previous federated strategies not only on the test data of sites selected for training, but also on the data of independent sites whose data was not used in the training process at all.

Related work

FL for medical image analysis

The study of FL in the area of medical image analysis is relatively new but it is well acknowledged as a promising solution for the existing data governance and privacy issues in the domain [18]. Following the first application of FL in the domain [19], FL approaches in the medical imaging domain literature have focused on various specific tasks including analysis of brain imaging data [19–26], CT hemorrhage segmentation [27], breast density estimation in mammography data [28, 29], pancreas segmentation in abdominal CT images [30], and tumor classification [31], histological sub-typing and survival prediction [32] and tumor-infiltrating lymphocytes (TILs) classification [33] on histopathology images.

There have also been efforts to create open frameworks and libraries to aid the development and employment of FL in the medical domain [34, 35]. Also, a challenge, namely The Federated Tumor Segmentation (FeTS) challenge [36], has been conducted on the task of segmenting brain tumors using multi-institutional magnetic resonance imaging (MRI) data.

Most of the FL studies in medical imaging have employed FedAvg method for model aggregation [19–26, 28–30, 32, 33]. Andreux et al. [31] proposed an enhancement for aggregation of batch normalization (BN) layers. Remedios et al. [27] incorporated momentum in the gradient averaging method.

Colorectal cancer (CRC)

The histopathology images used in this study comprise colorectal cancer (CRC) patients. Tumor classification and micro-satellite instability prediction are selected as use-case classification tasks to evaluate the performance of the proposed algorithm. Colorectal cancer (CRC) is the second leading cause of cancer related premature mortality worldwide with almost 10% of cancer related deaths of total 9.96 million. It is predicted to grow worldwide from an estimated 1.93 million in 2020 to 3.2 million in 2040 [37]. During the last couple of decades, the burden on the health system has increased because of increased screening for early detection.

The automatic diagnostic or prognostic systems by machine learning framework make use of routine histology images as input. However, the gigapixel whole-slide images (WSIs) of a standard glass slide, which can have 150,000×100,000 pixels, can't be processed in modern-day computers. The common workaround is dividing WSIs into small image tiles or patches, which become the input data of machine learning models for automatically learning the discriminative patterns. Several studies have been conducted to assist diagnosis and prognosis of CRC in the clinical practice [38–40]. The detection of tumor tiles has also been required as a preprocessing step which has significant impact on the performance of the downstream applications like CRC cancer screening [41] and the prediction of molecular pathways and microsatellite instability [39, 42].

Determining the molecular pathways of CRC patients is crucial for therapeutic decision-making. Microsatellite instability (MSI) is a molecular pathway associated with the genetic hypermutability of CRC tumors caused by dysfunction of DNA mismatch repair (MMR) genes. MSI tumors respond better to immunotherapy [43].

Materials and methods

In this section, we introduce the dataset and methods used in this study. An overall view of the proposed *FedDropoutAvg* method can be seen in Fig. 1.

The dataset

TCGA CRC-DX dataset

The dataset used in this study comprises of multi-gigapixel whole-slide images (WSIs) of 599 diagnostic slides from 591 colorectal cancer (CRC) patients contributed from 36 different sites to The Cancer Genome Atlas (TCGA) project. We used the Otsu thresholding [44] to extract tissue region from each

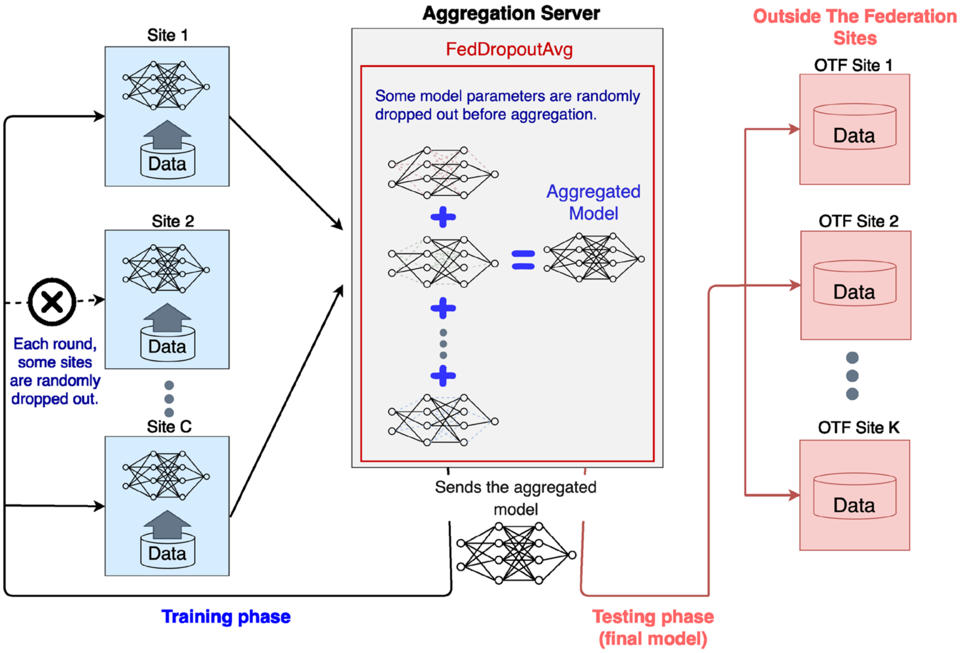

Site 1
Data
Site 2
Data
Each round, some sites are randomly dropped out.
Site C
Data
Aggregation Server
FedDropoutAvg
Some model parameters are randomly dropped out before aggregation.
Aggregated Model
Sends the aggregated model
Training phase
Testing phase (final model)
Outside The Federation Sites
OTF Site 1
Data
OTF Site 2
Data
OTF Site K
Data

Fig. 1 Concept diagram of the proposed *FedDropoutAvg* approach. **Training phase:** At the beginning of each training round, *Central Aggregation Server* sends the global consensus model to some sites which are randomly selected from all the sites participating in the training. Using the received model, local training takes place at each site on their local datasets. At the end of local training epochs, each *Site i* sends the parameters of their locally trained model to the *Central Aggregation Server*. Then, *Central Aggregation Server* randomly drops out some of the parameters of the received models and aggregates the models into a global consensus model by averaging. This training process continues for some number of rounds. **Testing phase:** After the federated training is over, the *Central Aggregation Server* sends the final consensus model to independent sites outside the federation for use on their own test sets, simulating a real-life scenario

slide. The non-overlapping square tiles of size 512 × 512 were extracted from the segmented tissue region at 20× magnification.

Tumor segmentation

For tumor segmentation, we fine-tuned ResNet18 [45] pre-trained on ImageNet to distinguish between tumor and non-tumor tiles of each slide for the FL experiments. A total of 35436 tiles are extracted from seven randomly picked TCGA slides and two publicly available data sets [39, 46]. Seventy percent of the data (24843 tiles) was split for training, fifteen percent each for validation (5380 tiles), and held-out test set (5213 tiles with 2493 non-tumor tiles and 2720 tumor tiles). The network distinguished the tiles of the unseen test set belonging to the tumor and non-tumor classes with an accuracy of 99% [42]. We used this trained network to separate the tumor and non-tumor tiles from the entire TCGA cohort.

Data collection for FL tumor classification experiments

Using the tiles and their labels as described above, we initially collected a multi-institutional dataset containing samples from 36 different sites. We have excluded sites contributing data of fewer than five patients from this study and randomly divided the dataset of the remaining 21 sites into federated training (11 sites, *Local Sets*) and independent test set (10 outside the federation (OTF) sites, *Independent Sets*). The data of *Local Sets* is patients-wise split into training, validation, and test sets (50%, 10%, 40%), keeping the tiles belonging to the same patient in the same set. Only the training set of *Local Sets* is used in model training and the validation set of *Local Sets* is used to select the best model parameters. In federated training, each local model is trained on the corresponding local training set of the sites in the *Local Sets*. In classical *Centralized* training, a single model is trained on the union of the training sets of the sites' data. Test sets of the sites in the *Local Sets* and all of the data of the OTF sites in the *Independent Sets* have been used for evaluation purposes. More details are shown in Table 1.

Data collection for MSI prediction experiments

Only the portion of the tiles labeled as tumor is used for the MSI prediction task along with corresponding patient-level MSI status labels. Out of total the 36 sites, only 19 of them have at least one MSI positive case. For this task, we have only used data from these 19 sites. We have constituted 4 groups (see Table 2) from these sites to perform 4-fold cross validation experiments.

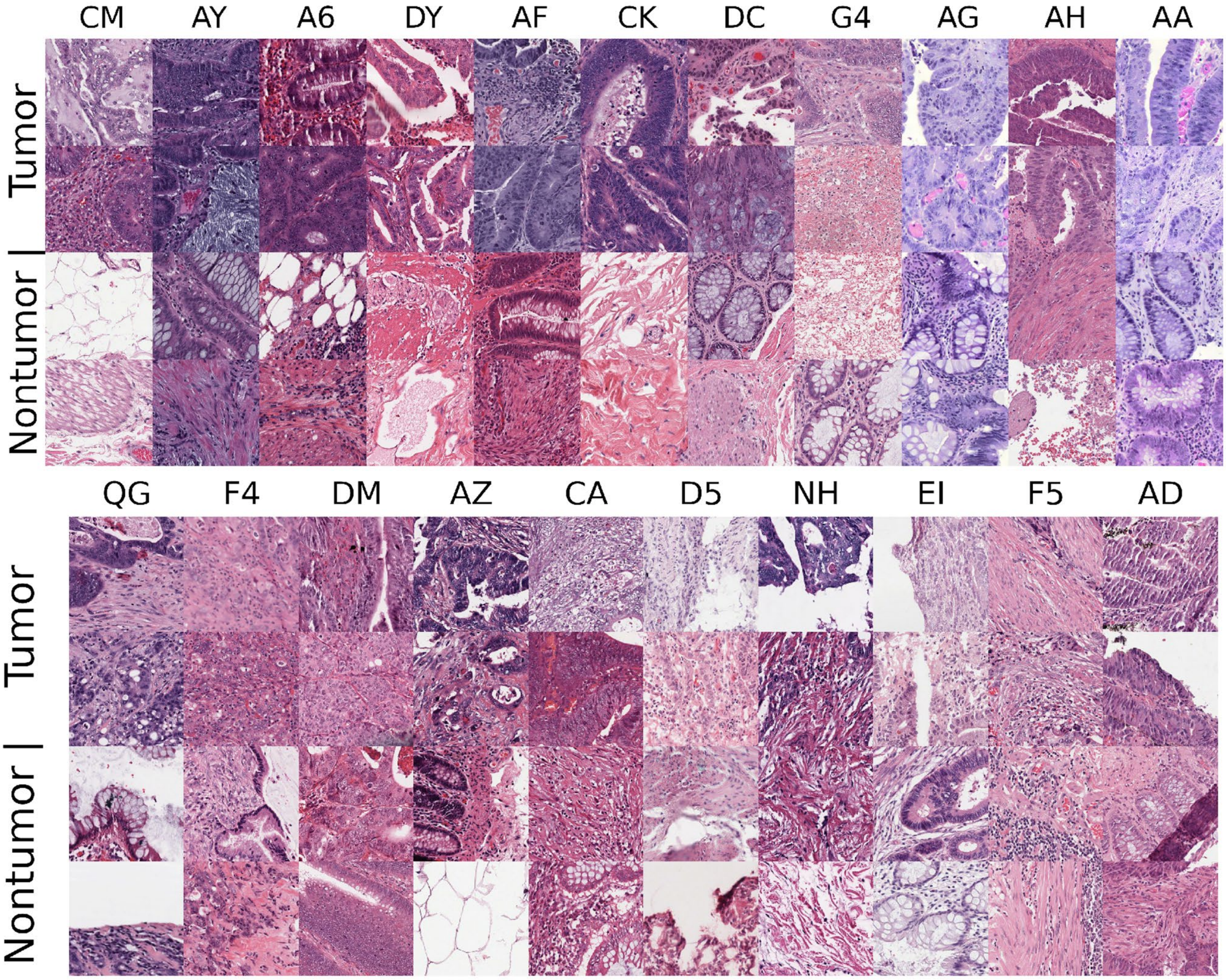


Fig. 2 Sample tumor and non-tumor image patches from different sites. It can be observed that there is a large intra-class variation between the two classes. Some image patches contain artifacts; stain color variation can also be seen in images from different sites

Federated dropout averaging (*FedDropoutAvg*)

Dropping out model parameters before aggregation

Our proposed method is based on FedAvg proposed by [3] which is the most popular method used for model aggregation in federated systems. In FedAvg method, in each federated round t, global consensus model parameters w^{t+1} are calculated as follows:

$$w^{t+1} = \sum_{k \in C} \alpha_k w_k^t \tag{1}$$

Here, α_k is the contribution weight of each client k in this weighted averaging (aggregation) equation.

In standard federated averaging, α_k is calculated as the proportion of number of data samples n_k of each client $k \in C$ to total number of samples of all clients participated in training N.

$$\alpha_k = \frac{n_k}{N} \quad \text{where} \quad N = \sum_{k \in C} n_k \tag{2}$$

In the *FedDropoutAvg* method, we propose to drop out some of the parameters from each client model w_k^t before aggregation and adjust the client contribution weights accordingly. Here, we define a new parameter *Federated Dropout Rate* (*fdr*), where $fdr = 0$ is the same as the standard FedAvg. At the end of each round, we create random masks for each client model w_k^t, and use those masks to select the model parameters (weights) which will be included in the aggregation process.

Table 1 Data used in tumor classification experiments. Number of patients (n), number of tumor (N^i_t) and non-tumor (N^i_{nt}) image patches (each patch being 512×512 pixels) per site (client i) and total number of patches (N) from the TCGA-CRC-DX dataset used in the training (TR), validation (VAL) and test (TS) sets. Training data from sites "CM" through "AA" used for training, their hold-out test sets and data from "QG" through "AD" used for testing

		All sites																				
		CM	AY	A6	DY	AF	CK	DC	G4	AG	AH	AA	QG	F4	DM	AZ	CA	D5	NH	EI	F5	AD
TR	n	18	5	24	3	9	7	6	13	36	3	78	2	8	11	10	5	15	4	8	6	6
	N^i_{nt}	53.5K	4.7K	13.9K	5.7K	6.9K	27.3K	15.2K	54.2K	8.5K	1.8K	30.9K	12.3K	23.7K	24.0K	31.0K	8.5K	33.9K	18.6K	26.8K	12.7K	2.1K
	N^i_t	27.0K	5.9K	11.9K	5.5K	5.4K	9.2K	11.2K	14.6K	19.4K	1.0K	46.0K	4.7K	12.0K	7.1K	13.4K	5.8K	9.6K	5.0K	7.1K	6.0K	2.5K
VAL	n	4	2	5	1	2	2	2	3	8	1	16	1	2	3	3	2	4	1	2	2	2
	N^i_{nt}	14.1K	0.2K	1.8K	1.5K	1.2K	8.1K	4.0K	9.9K	1.6K	0.2K	3.0K	2.3K	3.5K	6.9K	11.0K	3.1K	3.1K	3.2K	4.0K	3.8K	299
	N^i_t	6.8K	1.9K	2.3K	1.3K	0.3K	4.2K	5.6K	3.5K	4.3K	1.0K	11.4K	2.8K	3.2K	2.4K	3K	3.9K	1.7K	3.5K	399	3.5K	676
TS	n	15	3	19	2	7	5	5	11	28	3	62	2	6	9	7	3	12	4	7	4	5
	N^i_{nt}	21.7K	0.4K	5.3K	2.6K	2.0K	12.0K	12.2K	37.2K	1.9K	0.3K	4.0K	4.3K	7.1K	18.1K	19.4K	2.8K	5.9K	7.1K	6.9K	9.2K	653
	N^i_t	24.6K	4.1K	8.9K	1.4K	3.3K	8.6K	6.1K	16.6K	14.9K	1.7K	32.3K	1.6K	4.9K	6.1K	11.2K	4.7K	3.5K	6.2K	3.8K	3.7K	2.8K
N		147.7K	17.3K	44.1K	17.9K	19.1K	69.4K	54.3K	135.9K	50.1K	6.0K	127.6K	28.0K	54.4K	64.7K	89.0K	28.7K	57.7K	43.6K	49.0K	38.9K	9.0K

Table 2 Data used in MSI prediction experiments. The sites in each group, and the number of MSI positive cases and the number of all cases per group. In the 4-fold experiments, at each fold, data from 3 groups used for training while the remaining group is used for testing

Group No	sites in the group	(# pos.) / (# all)
Group 1	DC, NH, QG, WS, AD, A6, AZ	15 / 82
Group 2	AA	13 / 74
Group 3	CK, CM, D5	16 / 80
Group 4	EI, F4, G4, 5M, AG, AM, AU, AY	17 / 110

For a more formal explanation, let parameter $p^t_{i,k}$ be any parameter (weights or biases) at index i in model w^t_k. Then, p^{t+1}_i, the parameter at the same index of the aggregated global model will be calculated as follows:

$$p^{t+1}_k = \sum_{k \in C} \alpha^t_{i,k} p^t_{i,k} \tag{3}$$

where, $\alpha^t_{i,k}$ is the contribution weight of each client parameter $p^t_{i,k}$ and obtained as follows:

$$\alpha^t_{i,k} = \frac{n_k R^t_{i,k}}{N^t_i} \quad \text{where} \quad N^t_i = \sum_{k \in C} n_k R^t_{i,k} \tag{4}$$

Here, N^t_i indicates the total number of data samples of all the clients whose parameters at index i are not dropped out for the aggregation at the end of round t.

```
**Server executes:**
  initialize $w_0$
  $m \leftarrow (1 - \lambda_{cdr}) \cdot C$
  **for** each round $t = 1, 2, \ldots, T$ **do**
    $S_t \leftarrow$ (a random subset of m clients)
    **for** each client $k \in S_t$ **do**
      $w^k_{t+1} \leftarrow$ ClientUpdate($k, w_t$)
    initialize $w_{t+1}$
    **for** each parameter $p_{t+1} \in w_{t+1}$ **do**
      $S_u \leftarrow$ random subset of $S_t$, each client $k \in$
      $S_t$ is chosen with probability $(1 - \lambda_{fdr})$
      $N \leftarrow \sum_{k \in S_u} n_k$
      $p_{t+1} \leftarrow \sum_{k \in S_u} \frac{n_k}{N} p^k_{t+1}$

**ClientUpdate($k, w$):**   // on client k
  perform training procedure to update $w$ for $E$
  epochs of SGD
  send $w$ back to the server
```

Algorithm 1 FedDropoutAvg: C: clients, n_k: the number of training samples of each client $k \in C$, T: the number of federated rounds, E: the number of local epochs, λ_{fdr} and λ_{cdr}: federated dropout rate and client dropout rate

The $R^t_{i,k}$ value in the calculation is a random Boolean value (0 or 1) which is obtained using the newly defined *fdr*. Where RandomUniform() draws a value from a uniform distribution over the half-open interval [0,1).:

$$R^t_{i,k} = (RandomUniform() > fdr) \tag{5}$$

Client dropout

At each federated round, a random subset of clients is selected to participate in model training at that round. We have defined a new parameter *Client Dropout Rate* (*cdr*) which modifies the number of clients selected at each round. For a specific *cdr*, number of random clients selected at each round is constant. Please note that, for the case where $cdr = 0$, all the clients will participate at each training round.

Model

We have used ResNet18 [45] model with group normalization (GN) [47] layers instead of BN layers, as the GN layers are known to be more successful in decentralized machine learning settings [48].

Implementation and training

Experiments

For the tumor classification task, models are federatively or conventionally trained on the training data of sites (clients) in the *Local Sets*. More specifically, for evaluation, we have compared the proposed *FedDropoutAvg* method with the following approaches:

Centralized model: This model is trained in a conventional way on the union of data from all of the training sets.

Local models: Each one of these models is trained in a conventional way on data from each different training set.

FL approaches:

- *FedAvg* [3]: In this method, to obtain a global model, the server performs weighted averaging of all of the model parameters at each round.
- *FedProx* [5]: In this method, there is an additional term in the loss function which, during training, penalizes how much the model differs from the global model received at the start of the round. This term is adjusted by a parameter (μ). This parameter is selected as 0.01 after a grid search from 0.5, 0.1, 0.01, 0.001 based on the performance on the validation set.

Personalized FL (PFL) approaches: In these approaches, clients do not load all of the parameters from the global consensus model. Instead, for some specific parameter groups (e.g., layers, blocks) of the model, each client uses the values they had at the end of the previous round. In other words, these parameter groups are kept private at each client and do not get updated with a global model. In return, at the end of the federated training, each client's model is different. To compare these methods with the other approaches, for the clients selected for training, we used each client's resulting personalized model on their test set. To compare their performance on the held-out test data from completely different (independent) OTF sites, we have tested each client's personalized model at each independent center's data, and taken the average.

- *FedBN* [7]: In this approach, parameters of the batch normalization (BN) layers are kept private (personalized), and not updated using the global model parameters during training. ResNet18 model with BN layers is used.
- *PFL_s1-s4* [6, 8]: This model is trained by loading all of the layers, except the final fully connected layer, from the global model at the start of each round. For the naming, we label each block and stage in the model (ResNet18) used in the experiments using the same way with [8]. Specifically, the model has four stages (labeled with 's1','s2','s3' and 's4'), each with two ResBlocks (i.e, 's1' has ResBlocks 'A' and 'a','s2' has ResBlocks 'B' and 'b','s3' has ResBlocks 'C' and 'c','s1' has ResBlocks 'D' and 'd',) and a fully-connected layer. The model *PFL_s1-s4* is trained by loading stages 's1' to 's4' from the global model at the start of each round after the first one, keeping the final fully-connected layer private (personalized).
- *PFL_ABCD* [8]: Using the same labeling with the previous one, this model is trained by loading ResBlocks 'A','B','C' and 'D', keeping the others private.
- *PFL_abcd* [8] Similarly to the previous one, this model is trained by loading ResBlocks 'a','b','c' and 'd', keeping the others private.

FL using a model with dropout layers: This is done to examine the effect of using a dropout mechanism at the federated averaging process as proposed versus applying the standard FL approach using models having conventional dropout layers [14]. In this experiment, a dropout layer with a dropout rate r is added after every ReLu layer in the ResNet18 model. Different values of r (i.e, 0.1, 0.2, 0.3, 0.4, 0.5) are experimented. The best-performing r value on the validation set was 0.5.

All of the models are trained from scratch on GPU for each comparison. For the implementation of the models and methods, we used PyTorch. For model training, we used class-weighted binary cross-entropy loss and SGD optimizer with initial learning rate 0.1, momentum 0.9 and weight decay 0.0001, with the learning rate halved after every 2 epochs. Conventional training approaches (i.e, *local models* and *centralized model*) are trained for 20 epochs, while others are trained for 20 rounds (one epoch per round), and models from the epochs with the best validation loss have been selected. The proposed *FedDropoutAvg* model is same with *FedAvg* model when $cdr = 0$ and $fdr = 0$. For *FedDropoutAvg* model, the best values for cdr and fdr parameters are selected as 0.2 and 0.3, based on a grid search on $cdr \in [0, 0.1, 0.2, 0.4]$ and $fdr \in [0, 0.1, 0.2, 0.3, 0.4]$ on the validation set for the tumor classification task.

For the MSI prediction task, we have performed 4-fold cross-validation experiments. Sites are split into 4 different compute groups which act as different participants in the federated learning algorithms. At each fold, models are trained using all of the data of the corresponding split containing three groups (see Table 2), and they are tested on the remaining group which has not participated in training. For *FedDropoutAvg* model, since there are 3 groups that can participate in training at each round, we have fixed the cdr parameter in order to only drop one random group at each round. The other parameters and settings are the same with the tumor classification experiment.

Experimental results

In this section, we present the results of comparative analysis of our method with other FL methods, local training and centralized training approaches. We also analyze the effect of different cdr and fdr parameters on the performance of tumor classification task.

Tumor classification experiments

Limitations of local models

The F1 score of each locally trained model (rows) on each local test set is given as a heatmap in Fig. 3. Comparing federated models with locally trained models (Fig. 4), we observe that, on the individual local test sets, our proposed method, perform better on the test sets of 6 sites than the locally trained model of that site. This strengthens the motivation to put federated learning into practice.

Remarkably, although site A6 has an average number of samples compared to others, the locally trained model on this site's training set is the best generalizing model on the local test sets of all sites (with a mean F1 score of 0.781, see Fig. 3). Because of its smaller data size, this site contributes less in other federated and centralized learning approaches.

Comparing locally trained models with each other, we see that some of the locally trained models indeed do best on the test set of the same site they are trained on (CM, A6, DY, G4, AA). Surprisingly we observe this is not the case for the majority of them. For example, we see that the model locally trained on the training data of DC gives the best F1 results compared to other local models on the test sets of AY and AH. Likewise, among other local models, the locally trained model on A6 is best for AF, CK and DC; and the locally trained model on AA is best for AG.

These results point to the complexity of the underlying relationships between the different local models. They also support our initial motivation about not being able to measure the individual contributions of each local dataset, without sharing the datasets.

Experimental analysis

To compare different approaches, F1 scores and AUROC values are calculated for each comparison model on 21 different test sets (test data of training sites and data of independent sites).

In Table 3, the average performances of the centralized and federated approaches are shown. In the table, performance is given as the mean and the standard deviation of the metrics (i.e, F1, sensitivity, specificity, AUC) on the local and independent test sets data. As expected, the *Centralized* model, which is trained on all of the training data in a classical way, gives the best mean F1 score among all of the comparison models. After this, the proposed *FedDropoutAvg* model produced the most competitive results with the *Centralized* model.

In Fig. 4, we see the performance comparison of federated approaches on the individual local test sets (a) and for individual independent sets (b). In these results, we see that the proposed *FedDropoutAvg* method is consistently better than other FL and PFL approaches on these individual testing sets. This success of our proposed approach is attributed to the proposed client dropout and clients' model parameters drop-out mechanism. Our approach indeed helps avoid over-fitting to individual local datasets without requiring any information exchange among clients.

In Table 3 and Fig. 4 we see that the PFL methods show lower performance than the not personalized FL approaches including the proposed *FedDropoutAvg*. The PFL approaches involve training specific layers of a model locally on individual sites' datasets and keeping these layers private from server aggregation. The assumption is that these personalized models will perform better on their respective site's test datasets. However, the approach may result in overfitting on the training sets and perform poorly on unseen data. It may also not be as beneficial as the standard FL approach if the individual site's training data is not sufficient to train the personalized layers.

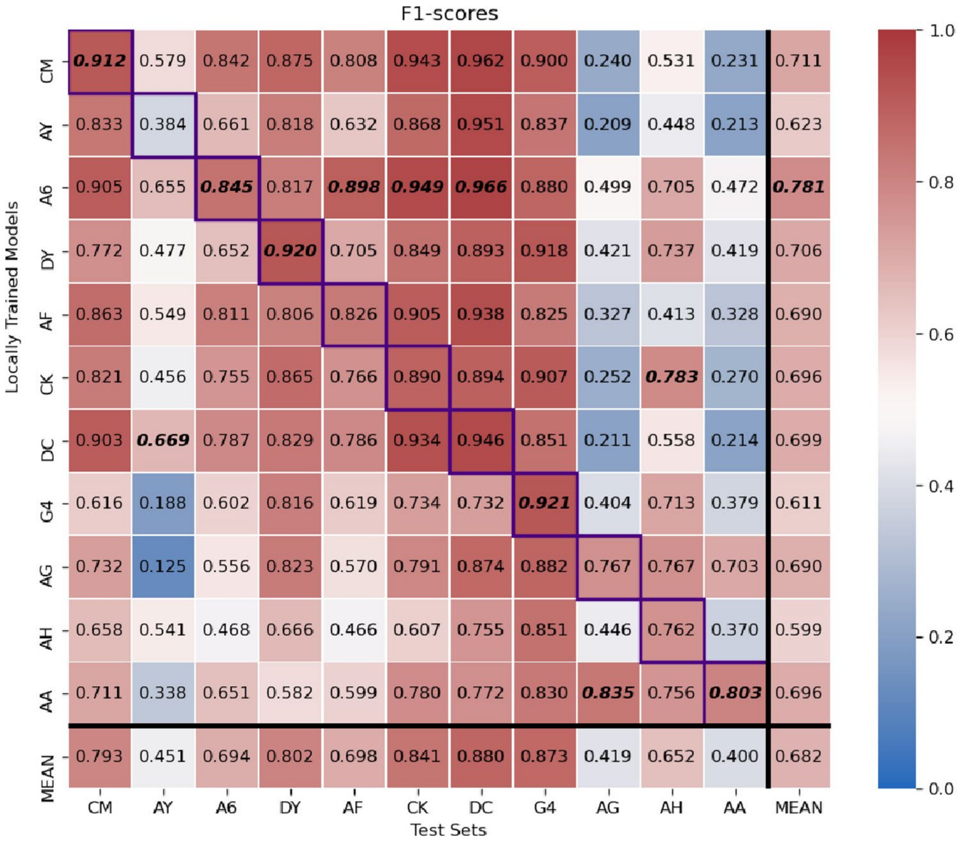
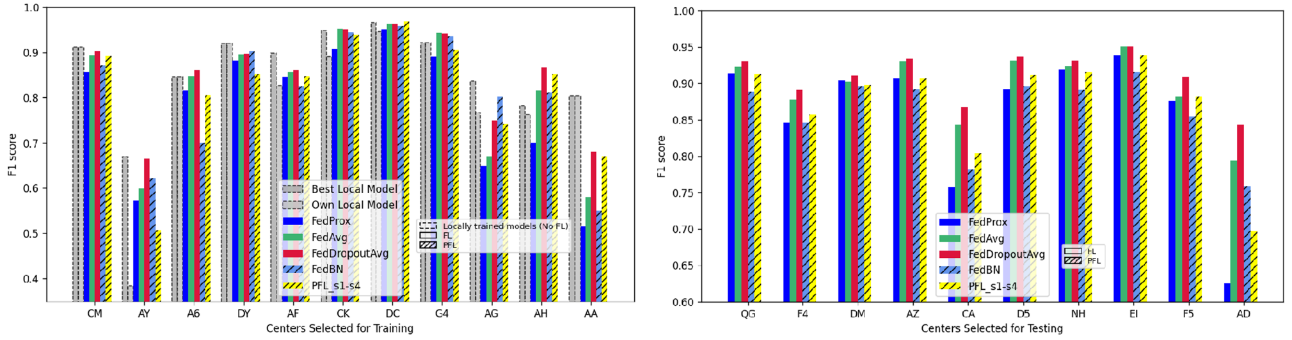


Fig. 3 Heatmap of F1 scores of the locally trained models (not federated) on the test sets. The mean F1 of each locally trained model is given on the rightmost column. Likewise, the mean F1 on each local test set give in the bottom row. F1 scores on the diagonal (highlighted with solid borders) correspond to F1 scores of the models trained on the training set and tested on the test set of the same site. The best F1 value on each test set (i.e., on each column) is given in bold

The FedProx shows very low performance on the independent test set 'AD' which is very imbalanced with the number of tumor samples being much higher than the number of non-tumor samples. Other test sets having a similar situation belong to the sites participating in training (i.e, 'AY', 'AG', 'AH', 'AA'') (Table 1). This could be the reason why the F1 performances of all of the federated approaches on the test sets of these sites are lower compared to other sites (Fig. 4). Although the performance is not as bad as 'AD', FedProx is still performing poorly on these. Most of the training sets, using which models are trained, have a balanced or higher number of positive samples compared to the



(a)

(b)

Fig. 4 F1 scores of the federated approaches on **a** the test sets of the sites selected for training, **b** the datasets of the sites selected for testing (independent OTF sites). The comparison methods (also see "Experiments") shown in this figure are as follows: 'Best Local Model', is the best performing model on that test set among all of the models locally trained on any site's training set (can also be seen in Fig. 3) and 'Own Local Model' is the model locally trained on the training set of that specific site. *FedAvg* and *FedProx* are classical FL methods. *FedDropoutAvg* is our proposed FL approach. Also, Personalized FL (PFL) approaches are shown: *FedBN* and *PFL_s1-s4*, which perform best compared to the other PFL ones

Table 3 Average performance of the centralized and federated approaches on the test sets of the sites selected for training and datasets of the independent out of the federation sites. Performance given as the mean and standard deviation of the metrics (F1, Sensitivity, Specificity and AUROC values). F1 scores on the individual datasets can be seen in the Figure 4

Methods		**Experimental Results**							
		Independent Test Sets				**Local Test Sets**			
		F1(±SD)	**Sens(±SD)**	**Spec(±SD)**	**AUC(±SD)**	**F1(SD)**	**Sens(±SD)**	**Spec(±SD)**	**AUC(±SD)**
Centralized Training		***0.939(0.021)***	***0.919(0.034)***	***0.933(0.031)***	***0.982(0.008)***	***0.915(0.056)***	***0.900(0.059)***	***0.969(0.021)***	***0.990(0.005)***
FL Methods	**FedAvg** [3]	0.896(0.046)	0.910(0.043)	0.793(0.081)	0.945(0.019)	0.819(0.133)	**0.874(0.094)**	0.912(0.053)	0.960(0.031)
	FedProx [5]	0.858(0.092)	0.921(0.071)	0.630(0.204)	0.923(0.042)	0.780(0.140)	0.819(0.152)	0.864(0.129)	0.936(0.040)
	FedDrAvg	**0.910(0.032)**	0.925(0.035)	**0.810(0.106)**	**0.954(0.018)**	**0.848(0.100)**	**0.881(0.101)**	**0.916(0.079)**	0.965(0.034)
PFL Methods	**FedBN** [7]	0.862(0.053)	**0.927(0.030)**	0.587(0.108)	0.914(0.023)	0.811(0.135)	0.828(0.166)	0.899(0.100)	**0.966(0.027)**
	pfl_s1-s4 [8]	0.872(0.072)	0.907(0.060)	0.708(0.154)	0.935(0.031)	0.816(0.133)	0.807(0.168)	0.903(0.095)	0.949(0.032)
	pfl_ABCD [8]	0.858(0.073)	0.899(0.052)	0.665(0.097)	0.907(0.028)	0.814(0.171)	0.854(0.075)	0.903(0.077)	0.955(0.045)
	pfl_abcd [8]	0.827(0.084)	0.885(0.040)	0.567(0.134)	0.862(0.042)	0.792(0.201)	0.858(0.113)	0.873(0.103)	0.932(0.097)

number of non-tumor samples. Although all of the models are trained using class-weighted cross-entropy loss to mitigate the class imbalance problem, the proximal term in the loss function formulation of the FedProx might be decreasing the positive effect of class-weighting. Our approach has better performance than the other FL and PFL models on these test sets.

In Fig. 6, qualitative results of different tumor classification methods on WSI level are presented with respective slide-level F1 scores. Centralized and FL methods are compared. The WSIs in this figure are from three different independent testing sites (AD, F4, NH), thus the models are not trained on these sites' datasets. For all of the WSIs, our proposed method has given better results than other FL approaches (FedAvg, FedProx) both qualitatively and in terms of F1 score.

In Table 4, comparison results of the experiments using *model with dropout layers* ("Experiments") trained with FedAvg and trained with the proposed *FedDropoutAvg* can be seen. Here, we see that the best-performing model with only *local dropout layers* without the proposed approach (i.e., 'ldr-0.1' which corresponds to ResNet18 models with local dropout layers with rates $r = 0.1$, trained with *FedAvg*) is not performing better than the proposed method without *local dropout layers* (*FedDropoutAvg* results in in Table 3). We also see that 'ldr-0.1' model trained with *FedDropoutAvg* performing better than 'ldr-0.1' trained with *FedAvg*. However, we do not see a benefit of adding dropout layers when using *FedDropoutAvg*.

Experimental analysis for training with different *cdr* and *fdr* parameters

To understand the effects of *cdr* and *fdr* parameters, we also trained the proposed *FedDropoutAvg* method with different parameters. In Fig. 5 tumor classification performance results of models trained with different *cdr* and *fdr* parameters can be seen. Keeping in mind that the model federatively trained

Table 4 Average performance of the experiments using *model with dropout layers* on the test sets of the sites selected for training and datasets of the independent out of the federation sites. 'ldr-0.1' to 'ldr-0.5' corresponds to ResNet18 models with local dropout layers (with rates *r* equal to 0.1 to 0.5, see "Experiments") trained with FedAvg. 'ldr-0.1 with FedDrAvg' corresponds to the 'ldr-0.1' trained with the proposed *FedDropoutAvg*. Performance given as the mean and standard deviation of the metrics (F1, Sensitivity, Specificity and AUROC values)

Model with Dropout Layers	**Independent Test Sets**				**Local Test Sets**			
	F1	**Sens**	**Spec**	**AUC**	**F1**	**Sens**	**Spec**	**AUC**
ldr-0.1 With FedDrAvg	**0.906(0.044)**	**0.947(0.025)**	**0.753(0.114)**	**0.954(0.018)**	**0.849(0.114)**	**0.878(0.096)**	**0.911(0.069)**	**0.965(0.029)**
ldr-0.1	0.889(0.057)	0.939(0.035)	0.709(0.130)	0.944(0.020)	0.828(0.109)	0.844(0.113)	0.891(0.104)	0.954(0.027)
ldr-0.2	0.881(0.060)	0.927(0.055)	0.690(0.152)	0.934(0.022)	0.807(0.125)	0.822(0.120)	0.888(0.119)	0.946(0.037)
ldr-0.3	0.881(0.055)	0.929(0.054)	0.677(0.152)	0.932(0.023)	0.820(0.121)	0.836(0.116)	0.897(0.098)	0.949(0.033)
ldr-0.4	0.878(0.057)	0.925(0.066)	0.664(0.177)	0.929(0.023)	0.819(0.118)	0.829(0.133)	0.896(0.099)	0.949(0.033)
ldr-0.5	0.849(0.089)	0.912(0.073)	0.598(0.211)	0.895(0.048)	0.773(0.143)	0.772(0.195)	0.854(0.186)	0.918(0.070)

with *cdr* = 0 and *fdr* = 0 corresponds to the FedAvg method in the literature, we see that selecting greater than zero values for both of the parameters provides gains in F1 score.

We can also interpret this figure as an ablation study. For example, if we compare the performance results of models trained with different *fdr* parameters when *cdr* parameter is equal to 0 (as in *FedAvg*) and equal to 0.2 (as the reported model here), we see that the models trained with *cdr* = 0.2 give close or better results than their counterparts (*cdr* = 0). Likewise, if we focus on the performance results of models trained with different *cdr* parameters when *fdr* parameter is equal to 0 (as in *FedAvg*) and equal to 0.3 (as the reported model here), we also see that the models trained with *fdr* = 0.3 give better results than their counterparts (*fdr* = 0).

Additionally, in Fig. 5, we can argue that in standard federated learning (i.e., when we do not use federated dropout, *fdr* = 0), if we decrease the number of clients participating at each round (i.e., increasing the client dropout rate, *cdr*) the performance does not differ noticeably.

MSI prediction experiments

We performed leave-one-group-out cross-validation experiments to evaluate the generalizability of all the methods on the data of unseen sites. Here, the sites are isolated to a single test fold, therefore if a patient's or site's data have been used in training it is not used in testing for that fold. At each fold, three groups have participated in training while the remaining group is used for testing. In the test time, patch level predictions are aggregated into patient-level predictions. For comparison we have used two different methods for aggregation: maximum pooling and proportion of positive class ('MSI-ness' [39]). For each fold, we have calculated the AUROC values on patient level predictions. The results of these experiments can be seen in table 5. These results show FL approaches, especially our proposed *FedDropoutAvg* method, performs better for MSI prediction and further confirm that our proposed approach is superior to other federated approaches in terms of average AUROC values on different test splits. The task of MSI prediction is a challenging problem since the MSI labels are available at slide-level and not at patch image level, meaning a slide may contain irrelevant or noisy patches that are not aligned with its label. We hypothesize that the federated models have a better ability to suppress noisy patch labels as compared to the centralized model.

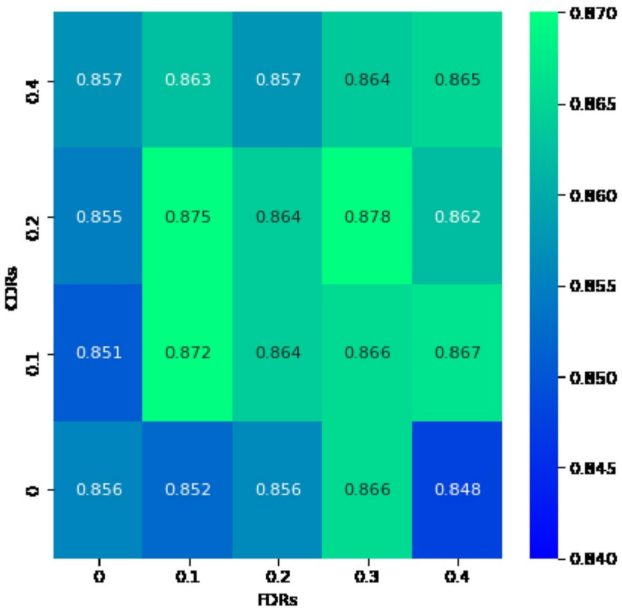


Fig. 5 Tumor classification performance results of proposed method trained with different *cdr* or *fdr* parameters. Performance given as the average of the F1 scores on all of the test sets for the proposed *FedDropoutAvg* method. For demonstration purposes here, performance is obtained on the test sets of the sites selected for training and datasets of the outside the federation sites selected for testing (not participated in training). The best parameters are originally selected on the validation sets of the training sites as *cdr* = 0.2 and *fdr* = 0.3. Note that *FedDropoutAvg* with *cdr* = 0 and *fdr* = 0 is the same with the *FedAvg* method from the literature

Discussion and future directions

In this study, we demonstrated *FedDropoutAvg* method as a better way to train models with federated learning for histopathological image analysis tasks. In the tumor classification application we presented, *FedDropoutAvg* achieved closer performance to the conventional training where all of the data is centralized in a data lake, compared to other major federated learning approaches. We think that our strategy will allow us to achieve the goal of training better and more robust models with higher clinical usefulness while maintaining the privacy of the data via federated learning.

The second application presented here, MSI prediction, is a WSI classification task. It is often termed multiple instance or weakly supervised learning problem. The MSI labels are available at slide-level but not at image patch levels, thus, a slide may contain irrelevant or redundant patches from the tumor regions. Table IV shows our proposed *FedDropoutAvg* method performs better for MSI prediction than centralized and federated learning methods. This shows dropping out clients and the parameters from locally trained models is helping model generalization. Additionally, the proposed dropout mechanisms seem to be a potential strategy for weakly supervised learning in federated or multicentric settings. In the future, we would like to extend our experiments to further understand the role of federated dropout mechanisms in the patch-level label noise suppression for different WSI classification tasks in a federated setting.

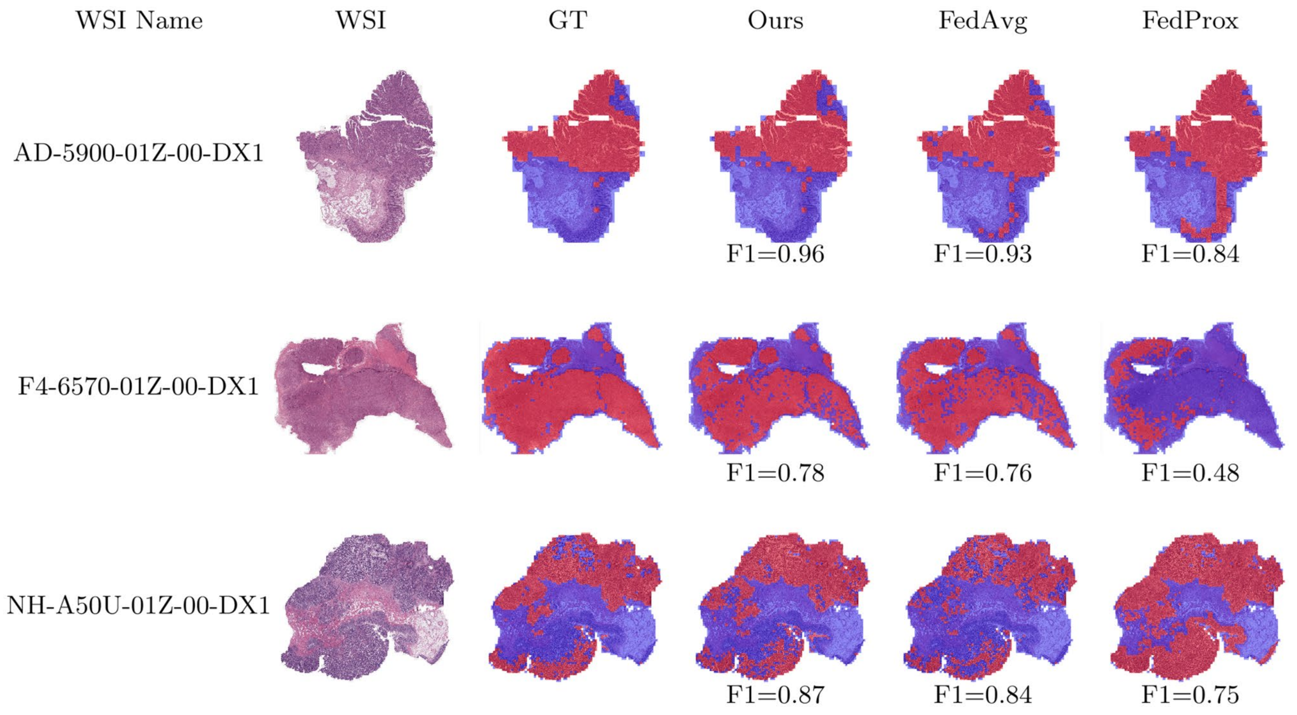


Fig. 6 Example qualitative results of different methods on WSI level. *First three columns*: WSI names, regions of interest (ROI) in WSIs from different independent sites (respectively: AD, F4, NH) and ground-truth segmentation masks. *Other columns*: binary segmentation masks of different methods overlaid on images, red areas indicating “tumor", blue areas indicating “nontumor" predictions. Respective slide-level F1 metrics are given below each result. All of the WSIs are from independent test sets and the presented methods are not trained on the datasets of the sites these WSIs belong to

The dropout method in the literature [14] is a generic technique to reduce over-fitting while training a routine neural network and it is different than our approach. it is to be used in a single neural network training and not for FL settings. In this technique, some parameters of a single network are dropped out randomly during training. The key difference between training a model with local dropout layers (as in [14]) and the proposed *FedDropoutAvg* is that our approach is proposed for adapting in the model aggregation step of a federated learning framework where multiple models are being trained at each round by different clients on different datasets and then, at the end of each round, some parameters of them are dropped out when aggregating (averaging) to obtain a global consensus model. This way the global model is not the result of every parameter of every client’s model. Whereas local dropout layers are used while training a single model on each batch of a single dataset, and even if we use them in FL, in the aggregation step all of the parameters of all of the models will still be averaged. While local dropout layers might help performance on a single dataset, in an FL framework, adding dropout layers to the model does not have an effect on the model aggregation step. The proposed method with or without *local dropout layers* (Tables 3 and 4) is better than only using a model with *local dropout layers*. However, in our experiments, we do not see an added benefit of having dropout layers in the model when using *FedDropoutAvg* compared to the performances of the ones without it.

Table 5 Leave-one-group-out MSI prediction experiments: average AUROC values of the centralized and federated approaches on the test sets at each fold. Performance given as the mean and standard deviation of the AUROC values as (Mean AUC (± SD)). AUROC values calculated for patient-wise MSI predictions obtained by two different methods (positive proportion and max pooling) aggregating patch-wise predictions

	Positive Proportion	MAX pooling
Centralized	0.6214 (± 0.0323)	0.6399 (± 0.0517)
FedAVG [3]	0.6782 (± 0.0868)	0.6573 (± 0.0746)
FedProx [5]	0.5820 (± 0.0217)	0.6376 (± 0.0869)
FedDrAVG	**0.6819 (± 0.0524)**	**0.6917 (± 0.0690)**

There are some studies under the name “Federated Dropout” [49, 50], however, these studies are different from our proposed “Federated Dropout Averaging” approach in which we propose to use dropout mechanisms for the model aggregation. Since those studies, “Federated Dropout” [50] and “Adaptive Federated Dropout” [49] are for FL in mobile devices, clients’ computational capabilities and communication efficiency is an important aspect due to the large number of devices involved in the training. The main goal of both of these approaches is to decrease client resource requirements and increase communication efficiency, by decreasing the model size to be trained, sent and received by the local clients(mobile devices). In these approaches, each local client trains a smaller model (a sub-model of the global model), while the server has the whole global model. Clients train a subset of the global model, and it is either a random subset [50] or a dynamically selected subset [49]. Then, the server maps and reunites those smaller locally trained models into the global model. Differently than these approaches, we propose to train the same whole global model architecture locally at each client, achieving better trained local models and a much more flexible drop-out application at aggregation time.

In this study, we only used the major FL optimization methods for comparison. We strongly believe that, in the future, our proposed approach can be enhanced by being combined with some of the other additional optimization techniques such as the gradient averaging with momentum technique by [27]).

As demonstrated, the *FedDropoutAvg* approach proposes to use lower number of clients which are randomly selected at each federated round. For example, when *cdr* = 0.2, only 8 random clients participate at each round out of 11 clients. As a result, at each round, training will only take place at the selected subset of clients. Even though there are less data being used at each round, it could produce good results compared to other federated approaches as presented. In Fig. 7, the total cross-entropy (CE) losses of the federated approaches on the training and validation sets at the end of each training round can be seen. Here we can see that convergence rates of the federated approaches do not differ substantially in our experiment. In the future, this aspect can be explored with detailed convergence analysis. Also, the benefits of this on communication efficiency and the total amount of computation time can be explored for a real-world histopathology image analysis system. The clients who are not selected do not need to communicate with the server at the end of the round, and this might be providing a way to increase communication efficiency. Also, total amount of computation time might be decreased, since it is proportional to the total amount of data of the selected clients. We did not provide an analysis of this aspect since the amount of data of each local training client is very heterogeneous in our dataset. In the future, additional multiple experiments can be done to understand the computational efficiency of the proposed method. We plan to do an extensive theoretical analysis of the proposed method in the future.

In Fig. 5, we see performance results (discussed in "Experimental Analysis for Training with Different *cdr* and *fdr* Parameters") of the proposed method trained with different *cdr* or *fdr* parameters. Due to the inherently stochastic nature of the dropout mechanisms, we see small fluctuations in the performance, however here we can say that selecting a *cdr* value between 0.1-0.2 and an *fdr* value between 0.1-0.3 seems like a sweet spot in our experimental settings.

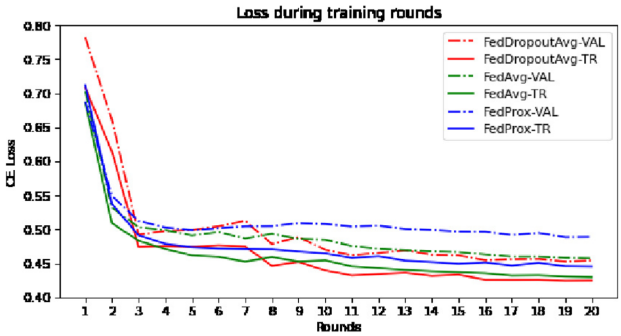


Fig. 7 CE loss vs communication rounds of the federated approaches. Average CE losses across the training and validation sets of the sites selected for training are given

We acknowledge that due to the client (center) dropout mechanism of the *FedDropoutAvg*, it might be difficult to deploy this method for a setting with a small number of participating sites with small datasets, which is usually the case for medical image analysis studies. To see if the proposed *FedDropoutAvg* method is still applicable when the number of training clients is lower, we changed our setting for the tumor classification experiments, to only select four sites (i.e., 'CM','CK','G4' and 'AA', the sites having the highest amount of training data) to participate in training, and adjusted the client dropout rate to 0.25, so one client will be dropped out randomly at each training round. The performance in terms of average AUC became 0.9392(±0.0296) on the independent sites and 0.9472(±0.0452) on the local test sets in the original experimental setting, showing it is still possible to train a model using *FedDropoutAvg* with lower number of clients. In the future, we want to thoroughly analyze the effects of the number of clients participating in training and the number of samples in their datasets.

Conclusions

Federated learning can help different institutions contribute to the training of powerful models without requiring any training data to be shared. In this paper, we proposed *FedDropoutAvg* and explored this federated training approach for real-world multi-site histopathology image classification and compared it with various existing federated learning methods. We evaluated the trained models on an independent test set of clients which have not participated in the training process. We showed that by using the proposed federated learning method, it is possible to achieve a classification performance comparable to a centralized model that requires access to data from all the clients used for training the model. To be more specific, we showed that our proposed approach performs 1%-9% better compared to other FL approaches and reduces the performance gap to less than 3% between FL and a central deep learning model, in terms of AUC on independent test sites.

Model aggregation is a critical piece of the FL paradigm and the improvement in performance and generalization ability of this new federated aggregation method has rich potential for usage in future FL models. In our experiments, we have demonstrated the effectiveness of our method on multi-gigapixel colon histology image data from 21 sites, comprising 1.2 million image patches (each patch of 512x512 pixels). In the future, we would like to explore the potential of our proposed method for other types of data and other histology image datasets.

In this study, we did not examine the privacy limitations of the proposed approach and we did not consider the data leakage from the model parameters if someone attempts to reconstruct the data using the model parameters exchanged during the federated training (i.e., a model inversion attack). In the future, the effects of the proposed approach on privacy and the combination of the proposed approach with different privacy-preserving techniques could be examined.

Author Contributions All authors conceptualized the study. GG was responsible for the drafting and final manuscript preparation, the software, the analysis of the results, and the preparation of figures and tables. NR designed the overall study setting with GG. MB contributed to data preparation. All analyzed, interpreted and helped to communicate the results. All provided significant writing, reviewing, and editing of all versions.

Funding This work was partly supported by the Department of Computer Science, University of Warwick, and partly by GSK.

Availability of Data and Materials This research study was conducted retrospectively using human subject data made available in open access by TCGA Research Network (https://www.cancer.gov/tcga). Ethical approval was not required as confirmed by the license attached with the open-access data.

Declarations

Competing Interests NR is a Director and CSO of Histofy Ltd. NR declares that he is in receipt of research funding from AstraZeneca and GlaxoSmithKline (GSK) and is also a GSK Chair of Computational Pathology at the University of Warwick, UK. All other authors declare no competing interests.

References

1. Litjens G, Kooi T, Bejnordi BE, et al (2017) A survey on deep learning in medical image analysis. Medical image analysis 42:60–88
2. Konečnỳ J, McMahan HB, Ramage D, et al (2016) Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527
3. McMahan B, Moore E, Ramage D, et al (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, PMLR, pp 1273–1282
4. Kaissis GA, Makowski MR, Rückert D, et al (2020) Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence 2(6):305–311
5. Li T, Sahu AK, Zaheer M, et al (2018) Federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127
6. Arivazhagan MG, Aggarwal V, Singh AK, et al (2019) Federated learning with personalization layers. arXiv preprint arXiv:1912.00818
7. Li X, Jiang M, Zhang X, et al (2021) Fedbn: Federated learning on non-iid features via local batch normalization. arXiv preprint arXiv:2102.07623
8. Sun B, Huo H, Yang Y, et al (2021) Partialfed: Cross-domain personalized federated learning via partial initialization. Advances in Neural Information Processing Systems 34:23,309–23,320
9. Tan AZ, Yu H, Cui L, et al (2022) Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems
10. Liu Q, Chen C, Qin J, et al (2021) Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1013–1023
11. Achille A, Soatto S (2018) Information dropout: Learning optimal representations through noisy computation. IEEE transactions on pattern analysis and machine intelligence 40(12):2897–2905
12. Baldi P, Sadowski P (2014) The dropout learning algorithm. Artificial intelligence 210:78–122
13. Hinton GE, Srivastava N, Krizhevsky A, et al (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580
14. Srivastava N, Hinton G, Krizhevsky A, et al (2014) Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15(1):1929–1958
15. Bonawitz K, Eichner H, Grieskamp W, et al (2019) Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046
16. Li T, Sahu AK, Talwalkar A, et al (2020a) Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine 37(3):50–60
17. Li X, Huang K, Yang W, et al (2019b) On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189
18. Rieke N, Hancox J, Li W, et al (2020) The future of digital health with federated learning. NPJ digital medicine 3(1):1–7
19. Sheller MJ, Reina GA, Edwards B, et al (2018) Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In: International MICCAI Brainlesion Workshop, Springer, pp 92–104
20. Li W, Milletarì F, Xu D, et al (2019a) Privacy-preserving federated brain tumour segmentation. In: International Workshop on Machine Learning in Medical Imaging, Springer, pp 133–141
21. Li X, Gu Y, Dvornek N, et al (2020b) Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results. Medical Image Analysis 65:101,765
22. Pati S, Baid U, Edwards B, et al (2022) Federated learning enables big data for rare cancer boundary detection. Nature communications 13(1):7346
23. Roy AG, Siddiqui S, Pölsterl S, et al (2019) Braintorrent: A peer-to-peer environment for decentralized federated learning. arXiv preprint arXiv:1905.06731
24. Sarhan MH, Navab N, Eslami A, et al (2020) On the fairness of privacy-preserving representations in medical applications. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, p 140–149

25. Sheller MJ, Edwards B, Reina GA, et al (2020) Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Scientific reports 10(1):12,598
26. Silva S, Altmann A, Gutman B, et al (2020) Fed-biomed: A general open-source frontend framework for federated learning in healthcare. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, p 201–210
27. Remedios SW, Butman JA, Landman BA, et al (2020) Federated gradient averaging for multi-site training with momentum-based optimizers. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, p 170–180
28. Muthukrishnan R, Heyler A, Katti K, et al (2022) Mammodl: mammographic breast density estimation using federated learning. arXiv preprint arXiv:2206.05575
29. Roth HR, Chang K, Singh P, et al (2020) Federated learning for breast density classification: A real-world implementation. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, p 181–191
30. Wang P, Shen C, Roth HR, et al (2020) Automated pancreas segmentation using multi-institutional collaborative deep learning. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, p 192–200
31. Andreux M, du Terrail JO, Beguier C, et al (2020) Siloed federated learning for multi-centric histopathology datasets. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, p 129–139
32. Lu MY, Kong D, Lipkova J, et al (2020) Federated learning for computational pathology on gigapixel whole slide images. arXiv preprint arXiv:2009.10190
33. Baid U, Pati S, Kurc TM, et al (2022) Federated learning for the classification of tumor infiltrating lymphocytes. arXiv preprint arXiv:2203.16622
34. Foley P, Sheller MJ, Edwards B, et al (2022) Openfl: the open federated learning library. Physics in Medicine & Biology 67(21):214,001
35. Karargyris A, Umeton R, Sheller MJ, et al (2021) Medperf: open benchmarking platform for medical artificial intelligence using federated evaluation. arXiv preprint arXiv:2110.01406
36. Pati S, Baid U, Zenk M, et al (2021) The federated tumor segmentation (fets) challenge. arXiv preprint arXiv:2105.05874
37. Xi Y, Xu P (2021) Global colorectal cancer burden in 2020 and projections to 2040. Translational Oncology 14(10):101,174
38. Bilal M, Tsang YW, Ali M, et al (2022) Ai based pre-screening of large bowel cancer via weakly supervised learning of colorectal biopsy histology images. medRxiv
39. Kather JN, Pearson AT, Halama N, et al (2019) Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer. Nature medicine 25(7):1054–1056
40. Skrede OJ, De Raedt S, Kleppe A, et al (2020) Deep learning for prediction of colorectal cancer outcome: a discovery and validation study. The Lancet 395(10221):350–360
41. Wang KS, Yu G, Xu C, et al (2021) Accurate diagnosis of colorectal cancer based on histopathology images using artificial intelligence. BMC medicine 19(1):1–12
42. Bilal M, Raza SEA, Azam A, et al (2021) Development and validation of a weakly supervised deep learning framework to predict the status of molecular pathways and key mutations in colorectal cancer from routine histology images: a retrospective study. The Lancet Digital Health 3(12):e763–e772
43. Sinicrope FA, Sargent DJ (2012) Molecular pathways: microsatellite instability in colorectal cancer: prognostic, predictive, and therapeutic implications. Clinical cancer research 18(6):1506–1512
44. Otsu N (1979) A Threshold Selection Method from Gray-level Histograms. IEEE Trans on Systems, Man and Cybernetics 9(1):62–66. https://doi.org/10.1109/TSMC.1979.4310076, http://dx.doi.org/10.1109/TSMC.1979.4310076
45. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp 770–778
46. Shaban M, Awan R, Fraz MM, et al (2020) Context-aware convolutional neural network for grading of colorectal cancer histology images. IEEE Trans on Med Imag pp 1–1. https://warwick.ac.uk/fac/sci/dcs/research/tia/data/extended_crc_grading/
47. Wu Y, He K (2018) Group normalization. In: Proc. of the European Conf. on Computer Vision (ECCV), pp 3–19
48. Hsieh K, Phanishayee A, Mutlu O, et al (2020) The non-iid data quagmire of decentralized machine learning. In: International Conference on Machine Learning, PMLR, pp 4387–4398
49. Bouacida N, Hou J, Zang H, et al (2021) Adaptive federated dropout: Improving communication efficiency and generalization for federated learning. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp 1–6, https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484526
50. Caldas S, Konečny J, McMahan HB, et al (2018) Expanding the reach of federated learning by reducing client resource requirements. arXiv preprint arXiv:1812.07210

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.