

Article

# Effectiveness of Decentralized Federated Learning Algorithms in Healthcare: A Case Study on Cancer Classification

Malliga Subramanian<sup>1</sup>, Vani Rajasekar<sup>1</sup>, Sathishkumar V. E.<sup>2,\*</sup>, Kogilavani Shanmugavadivel<sup>1</sup> and P. S. Nandhini<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering, Kongu Engineering College, Erode 638060, Tamil Nadu, India

<sup>2</sup> Department of Industrial Engineering, Hanyang University, 222 Wangsimini-ro, Seongdong-gu, Seoul 04763, Republic of Korea

\* Correspondence: sathishkumar@hanyang.ac.kr

**Abstract:** Deep learning-based medical image analysis is an effective and precise method for identifying various cancer types. However, due to concerns over patient privacy, sharing diagnostic images across medical facilities is typically not permitted. Federated learning (FL) tries to construct a shared model across dispersed clients under such privacy-preserving constraints. Although there is a good chance of success, dealing with non-IID (non-independent and identical distribution) client data, which is a typical circumstance in real-world FL tasks, is still difficult for FL. We use two FL algorithms, FedAvg and FedProx, to manage client heterogeneity and non-IID data in a federated setting. A heterogeneous data split of the cancer datasets with three different forms of cancer—cervical, lung, and colon—is used to validate the efficacy of the FL. In addition, since hyperparameter optimization presents new difficulties in an FL setting, we also examine the impact of various hyperparameter values. We use Bayesian optimization to fine-tune the hyperparameters and identify the appropriate values in order to increase performance. Furthermore, we investigate the hyperparameter optimization in both local and global models of the FL environment. Through a series of experiments, we find that FedProx outperforms FedAvg in scenarios with significant levels of heterogeneity.

**Keywords:** federated learning; non-IID; heterogeneity; Bayesian optimization; FedAvg; FedProx



**Citation:** Subramanian, M.; Rajasekar, V.; V. E., S.; Shanmugavadivel, K.; Nandhini, P.S. Effectiveness of Decentralized Federated Learning Algorithms in Healthcare: A Case Study on Cancer Classification. *Electronics* **2022**, *11*, 4117. <https://doi.org/10.3390/electronics11244117>

Academic Editor: Gemma Piella

Received: 9 November 2022

Accepted: 9 December 2022

Published: 10 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recent technological improvements have altered intelligent healthcare powered by artificial intelligence (AI). Nevertheless, AI systems necessitate centralized data gathering and processing, which may be impractical in realistic healthcare contexts due to the scalability of modern healthcare networks and the need for data privacy. Federated Learning (FL), an emerging distributed collaborative AI paradigm, is particularly suitable for smart healthcare by coordinating the training of numerous clients, that is, in healthcare institutes, without the exchange of private data [1]. Deep learning, an AI technique, has proven its significant potential in medical image analysis for the early detection of chronic diseases by managing large amounts of health data to facilitate the delivery of healthcare services [2].

Smart healthcare systems have historically relied on cloud-based centralized AI functions for health data analytics. In light of the ever-increasing quantities of health data, this centralized system is inefficient because of the inherent delays in transmitting raw data, and the centralized system cannot accommodate substantial growth in network traffic. Furthermore, depending on a centralized server creates severe privacy risks [3]. This is of paramount importance, particularly in e-healthcare, where patient records are considered extremely confidential and protected by laws. In order to overcome the issues of data governance and privacy, FL trains algorithms cooperatively without exchanging the data itself. Compared to conventional machine learning methods, FL offers greater advantages [4] as listed below:

- i. Data diversity: heterogeneous data from multiple clients;
- ii. Real-time learning: models are continuously updated with data from clients;
- iii. Data security: the data are not sent to the central server and are kept only by the clients;
- iv. Hardware efficiency: FL does not need a complex central server to analyze.

Multiple clients work together using FL to tackle a machine-learning challenge. The central server builds and improves its model (global). This model is trained by the updates it receives from the clients. During training, the clients fine-tune local models using data they have. Instead of bringing data to the central server, like in traditional deep learning, the central server maintains a global shared model. Each client has its local model and sends the local model's weight or error gradient to the server. The central server then updates the global model based on the performance of the local models. This method iterates until the global model is trained.

### 1.1. Applications of FL

FL has numerous business applications, including traffic prediction and monitoring, healthcare, communication, IoT, transportation, autonomous vehicles, medical AI, and more [5]. In all of these applications, FL facilitates the collaborative learning of several devices utilizing a shared model. Thus, utilizing information from a wide variety of sources makes FL a practical tool. Once gathered, these data are used to update the model. These devices will only transmit the information collected from the local model updates to the server. Thus, FL is used where a secure transmission of private information is necessary. Below, we present a few domains where FL finds its application.

#### FL in healthcare

Because HIPAA and other rules make it difficult to share sensitive health information, healthcare is one of the industries that can gain the most from federated learning. A significant amount of data from diverse healthcare databases and devices can be used to develop AI models that adhere to regulations.

#### FL in Autonomous Vehicles

Because of its ability to make accurate predictions in real-time, FL is used in the development of autonomous vehicles. Information about current road and traffic circumstances might be added in real-time, facilitating faster learning and better decision-making. This could lead to a safer and more satisfying experience with autonomous vehicles. Hence, the automotive industry has great potential for the application of FL. However, at the present time, nothing but studies are being done in this area. These studies have shown that training time for predicting the angle at which a vehicle's wheels would turn when autonomously driving may be reduced through the use of FL.

#### FL in IoT:

For the development of smart and privacy-enhanced IoT systems, the notion of FL has recently been proposed [6]. For data training, FL is a distributed collaborative AI approach [7] that coordinates several devices with a central server without sharing actual datasets. To train a neural network, for instance, an intelligent IoT network may employ a group of IoT devices to act as "workers" communicating with a central server.

### 1.2. Our Approach

In this attempt, we use two FL algorithms to classify three types of cancer, namely cervical, lung, and colon cancer, and consider ten subclasses of these three types. The FL environment has an aggregation server and ten clients, each client has a local model that classifies each subclass of three types of cancer. The local model updates local weight parameters upon receiving updates from the global model, which runs on the aggregation server. The local model executes the FedAvg and FedProx algorithms to update their weight parameters and sends the updated weights to the aggregation server. The server aggregates the local weights and sends back the updated weight to the local models for further updates. This process is iterated until convergence. To tune hyperparameters in

both clients and the server, we use Bayesian optimization search to find appropriate values of the hyperparameters. Further, the local models send the local updates to the server if only it finds an improvement in performance.

### 1.3. Research Focus and Contributions

The purpose of this research is to evaluate the performance of FL algorithms against non-IID datasets. To enhance the performance, we use hyperparameter optimization techniques in both local and global models. As the number of communication rounds between the server and the clients, and the training methodology play a significant role in performance, we plan to address the following questions:

RQ1. Can we reduce communications between server and client models by tuning a hyperparameter locally before the federation process in the context of FL for a non-IID classification task?

RQ2. How much longer would the training take if the global model sent updates to every client instead of just a few clients who are chosen at random?

RQ3. What effect do various FL algorithms have on performance?

To address these research questions, we developed an FL framework that classifies non-IID datasets. For the development of the framework, the following contributions were made:

1. We applied the FedAVg and FedProx algorithms to different model architectures (MobileNet-V3, Xception-V3 and ResNet201) and presented the result of extensive experiments on three major types of cancer.
2. We used non-IID datasets to compare and evaluate the performance of FedAvg and FedProx algorithms.
3. We used Bayesian optimization to tune the hyperparameters of both local and global models to achieve better convergence.
4. We conducted numerous experiments using various computer vision tasks having different types of cancer.

To the best of our knowledge, FL has not yet been applied to classifying multiple types of cancer. The unique contribution is that the hyperparameter tuning process is implemented at both the server and the clients. Further, a client will send its update to the server only if it shows improved performance.

This article is further organized as follows: Section 2 presents a detailed report on the present FL research attempts. The proposed methodology along with the procedures in both client and server is described in Section 3. The experimental settings are recapitulated in Section 4. The results of extensive experiments are presented in Section 5, and this section also reports on the findings from these experiments.

## 2. Review of Literature

In the past decade, large-scale machine learning, especially in data center settings, has prompted the development of several distributed optimization algorithms [8]. As the power and popularity of electronic gadgets such as wearable devices, smart mobile phones, etc., increase, it becomes increasingly attractive to learn statistical models locally in networks of distributed devices, as opposed to sending the data to a central server. Such learning addresses the challenges of privacy issues, the heterogeneous nature of data and devices, and the distributed environment. This section examines current FL research efforts and emphasizes the contributions made by each.

### *IID vs. Non-IID Datasets*

In general, real-world data are non-independent and identically distributed (non-IID). In contrast, the majority of existing analytical and machine-learning techniques rely on IID data. Therefore, a suitable method is required to manage such types of real-world datasets. When a sample is said to be identically distributed, there are no overarching patterns, no fluctuations in the distribution, and each item is drawn from the same probability

distribution. Independent means that the sample items are all independent events. In other words, there is no relationship between them at all. Each subsample differs substantially and fails to accurately represent the dataset's underlying distribution if the data is non-IID.

The performance of the global model on non-IID federated data has been shown to be substantially poorer than that of IID [9–11]. Most of the present FL techniques frequently merely ignore the distribution divergence. The performance fairness is also made worse by the distribution divergence of non-IID data [12], which means that the established model produces noticeably varied local performance for various clients. This encourages us to use a non-IID FL approach by coordinating the local optimization goals for many clients. FL lets AI models train using private information without compromising privacy and helps to achieve better results in the healthcare field when there is a scarcity of data. It has paved the way for a lot of research [13].

He et al. [6] reconstructed FL as the FedGKT, a group knowledge transfer training algorithm, to train small CNNs on edge nodes and transfer their information through to a central server. FedGKT devises a variation of the alternating minimization technique. The authors used three different datasets (CIFAR-10, CIFAR-100, and CINIC-10) and their non-IID variations to train CNNs based on ResNet-56 and ResNet-110. Their findings indicate that FedGKT can achieve equivalent or perhaps slightly greater precision than FedAvg. A few studies have also attempted to investigate FL's benefits, drawbacks, and potential uses. The benefits and drawbacks of FL in IoT systems, as well as the ways in which it paves the way for a wide range of IoT applications, were explored by Zhang et al. [14]. Specifically, the authors presented seven major obstacles to FL in IoT platforms and discussed some new, promising solutions to these problems. In [2], the authors summarized the recent improvements of FL learning towards enabling FL-powered IoT applications, and they employed a set of measures, including sparsification, robustness, quantization, scalability, security, and privacy, to rigorously evaluate these developments. In addition to outlining various research questions and potential answers, [2] also developed a taxonomy for FL across IoT networks.

For non-IID FL, Zhang et al. [15] presented unified feature learning and an optimization objectives alignment approach (FedUFO). Extensive trials showed that FedUFO is superior to state-of-the-art techniques, including the competitive one data-sharing method. Sahu et al. [8] presented a system called FedProx to handle heterogeneity in federated networks in their study. According to them, FedProx may be thought of as a generalization and reparameterization of FedAvg. The authors have shown that, across several realistic federated datasets, FedProx enables more robust convergence than FedAvg. In order to better understand the effects of various hyperparameter optimization strategies in an FL system, Holly et al. [16] conducted extensive research. In an effort to reduce the overhead, the authors looked into a local hyperparameter optimization strategy that, unlike a global one, provides each client control over its own hyperparameter setting. The algorithms were tested on the MNIST data set with an IID partition and on an Internet of Things sensor-based industrial data set with a non-IID partition to determine how well they performed.

In [17], the authors described Auto-FedRL, a reinforcement learning-based federated hyperparameter optimization approach, where an online agent may dynamically alter the hyperparameters of each client depending on the current training status. Multiple search techniques and agents have been examined through extensive tests. Validation was performed on two real-world medical image segmentation datasets, one for COVID-19 lesion segmentation in chest CT and one for pancreas segmentation in abdominal CT, to ensure the effectiveness of the proposed technique.

Wang et al. [6] paid special attention to image processing applications that ensure the confidentiality of training data. Autonomous diagnosis of COVID-19 using deep learning algorithms has lately been a subject of research by scientists all around the world. It is true that several data-driven deep-learning approaches have been developed to make COVID-19 identification easier, but these data are still scarce due to patient privacy concerns. Under

this circumstance, the solution is FL, which allows many organizations to work together to create a powerful deep-learning model without having to share sensitive data.

Zhu et al. [18] analyzed the impact of non-IID data on parametric and non-parametric machine learning models for use in horizontal and vertical FL in great detail. Both the pros and cons of existing methods for dealing with the difficulties posed by non-IID data in federated learning were examined. In addition, several studies on detecting COVID-19 using FL were found throughout the current literature analysis [19–24]. These studies proved that an FL approach may successfully detect COVID-19-related CT abnormalities using external validation on patients from various healthcare institutes.

Using differential privacy, Beguiere et al. [25] suggested a method for FL training of genomic cancer prediction models. They showed that it is possible to achieve a good trade-off between accuracy and privacy by training a supervised model for the prediction of breast cancer using genomic data, which is split across two virtual centers while preserving data privacy with regard to model transfer through DP. Choudary et al. [26] presented an FL framework capable of learning a centralized model using decentralized health data stored in many places. There are two tiers of privacy in this framework. For one, it does not involve any data transfer or central server use when training the model. As a second line of defense against privacy breaches, the model employs a differential privacy mechanism. The authors proved that FL architecture might increase privacy while retaining global model's usefulness. A stratified Cox model was created by Hansen et al. [27] and validated at three different locations in different countries using FL. Claiming their models are well-calibrated with high power of discrimination, they demonstrated exceptional overall survival prediction for patients with laryngeal cancer who were treated with curative radiotherapy in three separate cohorts.

Furthermore, we also reviewed a few recent attempts based on FL for health cases. A new federated learning framework was proposed [28] that can develop predictive models through peer-to-peer collaboration without raw data exchanges, and this work uses a distributed algorithm to solve a binary supervised classification issue to predict hospitalizations for cardiac events. Kumar and Singla [29] demonstrated how well FL approaches may be adapted to the healthcare system and how they function, particularly in the areas of medication discovery, clinical diagnostics, digital health monitoring, and various illness forecasts and detection systems. This work [29] also includes comparative analysis, which compares various FL algorithms for various health sectors using metrics including accuracy, the area under the curve, precision, recall, and F-score. In an attempt by [30], the way in which FL can facilitate the growth of an open health ecosystem with the use of AI was described. This work also covers current issues with FL and potential remedies. Antunes et al. [31] presented a systematic review of the research on FL in the context of an electronic health record for healthcare applications. In their analysis, they highlighted the most important research topics, proposed solutions, case studies, and ML methods. The article also presented a general architecture for applying FL to healthcare data.

To summarize, from the literature survey, we understand that only very few attempts have been carried out using non-IID data in an FL environment, and these works have focused on COVID-19 detection only. These works update the global models irrespective of the local model result. Thus, in the current research attempt, we use three types of cancer that are distributed across the clients in a non-IID fashion and implement FL algorithms. In the proposed work, a client updates the weights to the global model if the current local model parameters provide improved performance.

### 3. Proposed Methodology

In this section, we present FL approaches to diagnosing cervical, lung, and colon cancer from CT, and MRI scan images. FL solves problems with data collection and aggregation by only sharing model updates [8]. In this paper, we improve data security by splitting a model into two parts. We chose a natural split for CNN architectures: a shared part with general information about the feature domain and an unshared part with information

about the user's task. Here, the shared part represents the aggregation server, and the clients have their own private, unshared data. With such a split, important issues such as data confidentiality, access rights, data protection, and access to heterogeneous data can be easily addressed. Table 1 presents the notations used in the article with short descriptions.

**Table 1.** Terminologies used in the work.

Notation	Descriptions
$W^g_{(t)}$	Global weight parameter at time 't'
$W^l_{(t-1)}$	Local weight parameters received by a client at time 't - 1'
$W^l_{(t),\omega}$	Local weight parameters updated by a client at time 't'
$N$	Number of clients
$l$	A local model
$E$	Number of epochs
$F_l$	Loss function
$\mu$	Proximal term

FL is a technique for distributed and secure deep learning that permits the construction of a shared model without compromising the confidentiality of the underlying data. An FL system is made up of two basic parts, the first of which is a local model that is executed on each client and the second of which is a global model that is executed on an aggregate server. In the following sections, a comprehensive explanation of both models is presented.

### 3.1. Local (Client) Models

The procedure adopted in client units is as follows:

```

LocalModelUpdate(N,l, Wg(t))://Run at each client 'N'
Wl(t-1) ← Wg(t) / Receive Wl(t-1) from the central server
For each epoch 1 to E
  For each batch 'e' of images
    Take a batch of images 'e' from the local dataset;
    Tune the hyperparameters using the Bayesian optimization search
    Train the CNN model, find the weight updates as ΔWl(t-1)
    Calculate the accuracy (LMUacc)
    Find the loss function (Fl) using the federated learning algorithms (FedAvg and FedProx)
    Update the weights, Wl(t) ← Wl(t-1) ± ΔWl(t-1)
    Copy ω ← Wl(t) (for FedProx algorithm)
  If LMUacc is greater than the accuracy before training, then
    Send the updated weights, Wl(t), to the server

```

### 3.2. Global (Server) Model

For training the global model, we used the following procedure.

```

Initialize global model: Wg(0)
t = 0
Repeat until performance is satisfied
{
  For each local model 'l' in N
    Wl(t+1) ← LocalModelUpdate(N,l, Wg(t))
    Tune the hyperparameters using the Bayesian optimization search
    Evaluate the global model using test images
    Update the weights (Wg(t)), if needed
    Wg(t+1) ← Wg(t) ± (1/N) summation (Wl(t+1))
    Send the updated weights Wg(t+1) to the clients
  t = t + 1
}

```

### 3.3. Federated Learning Schemes

We present two popular FL strategies in this section [2,8,32,33]. The difficulties in addressing the data heterogeneity among the datasets employed in the present work are taken into account by these two systems in distinct ways. Unbalanced local datasets and non-IID data among end devices are examples of data heterogeneity. The federated learning algorithms are explained below:

#### FedAvg

Federated averaging (FedAvg) is the most common and easiest way to aggregate data. There are several rounds of learning. At the end of each round, the server sends the current global model updates to each client. The clients then update the parameters of their local copy of the model by using SGD for several epochs to optimize the loss ( $F_l$ ) on their local training data. At the end of the round, all of the local parameters are sent to the server, which uses a weighted average to combine them. When all the parameters are added together, they make up the global model for the next round. The steps are enumerated below:

For each client  $n \in N$ ,  
 For epochs in  $E$   
     Find the minimal loss function  $F_l$  using the learning rate ( $\eta$ )  
      $W^l(t) \leftarrow W^l(t-1) + \Delta W^l(t-1)$  // Update  $W^l(t)$  using Stochastic Gradient Decent  
     Send the updated  $W^l(t)$  to the aggregation server

#### FedProx

As an alternative, FedProx is a generalization of FedAvg with minor tweaks to account for data and system heterogeneity. Once again, the learning occurs in cycles. The server updates the clients with the latest global model at the beginning of each round. In contrast to FedAvg, clients optimize a regularized loss containing a proximal term. To be more specific, the new function to minimize is  $F_l(\omega) + (\mu/2) (\|\omega - W^l(t-1)\|)^2$ , where  $F_l$  is the loss,  $\omega$  are the local parameters to optimize, and  $W^l(t-1)$  is the global parameters at time  $t$ . This strategy requires that local settings be uploaded to a central location, where they can be averaged with data from other users.

For each client  $n \in N$ ,  
 For epochs in  $E$   
     Compute the loss function  $F_l$  using the objective function as,  
      $F_l(\omega) + (\mu/2) (\|\omega - W^l(t-1)\|)^2$   
      $W^l(t) \leftarrow W^l(t-1) + \Delta W^l(t-1)$   
     Send the updated  $W^l(t)$  to the aggregation server

Each client trains their model independently in FL. In other words, each client undergoes a unique model training procedure. To integrate and feed the aggregated main model, only newly learned model parameters are provided to the aggregation server. Then, the server sends the combined global model back to all the clients, and this process is repeated. All of these parameters differ from one another since the global model's parameters and the parameters of every local model inside each client are randomly initialized. The entire process of FL is enumerated below:

1. Before local models in the nodes are trained, the global model communicates its parameters to the nodes.
2. Using these settings, client nodes begin to train their own local models on their own data.
3. While building its local model, each client fine-tunes the parameters using a Bayesian search and updates it.
4. Each node delivers its local model parameters to the global model after updating the local parameters.
5. The global model calculates its new weight parameters by averaging these values, then sends them back to the clients for the subsequent iteration.
6. To enhance the FL model's performance, repeat steps 1 through 5.

Figure 1 shows the proposed architecture. As can be seen from Figure 1, there are ten clients each with a dataset for one of the subtypes of cervical, lung, and colon cancer.

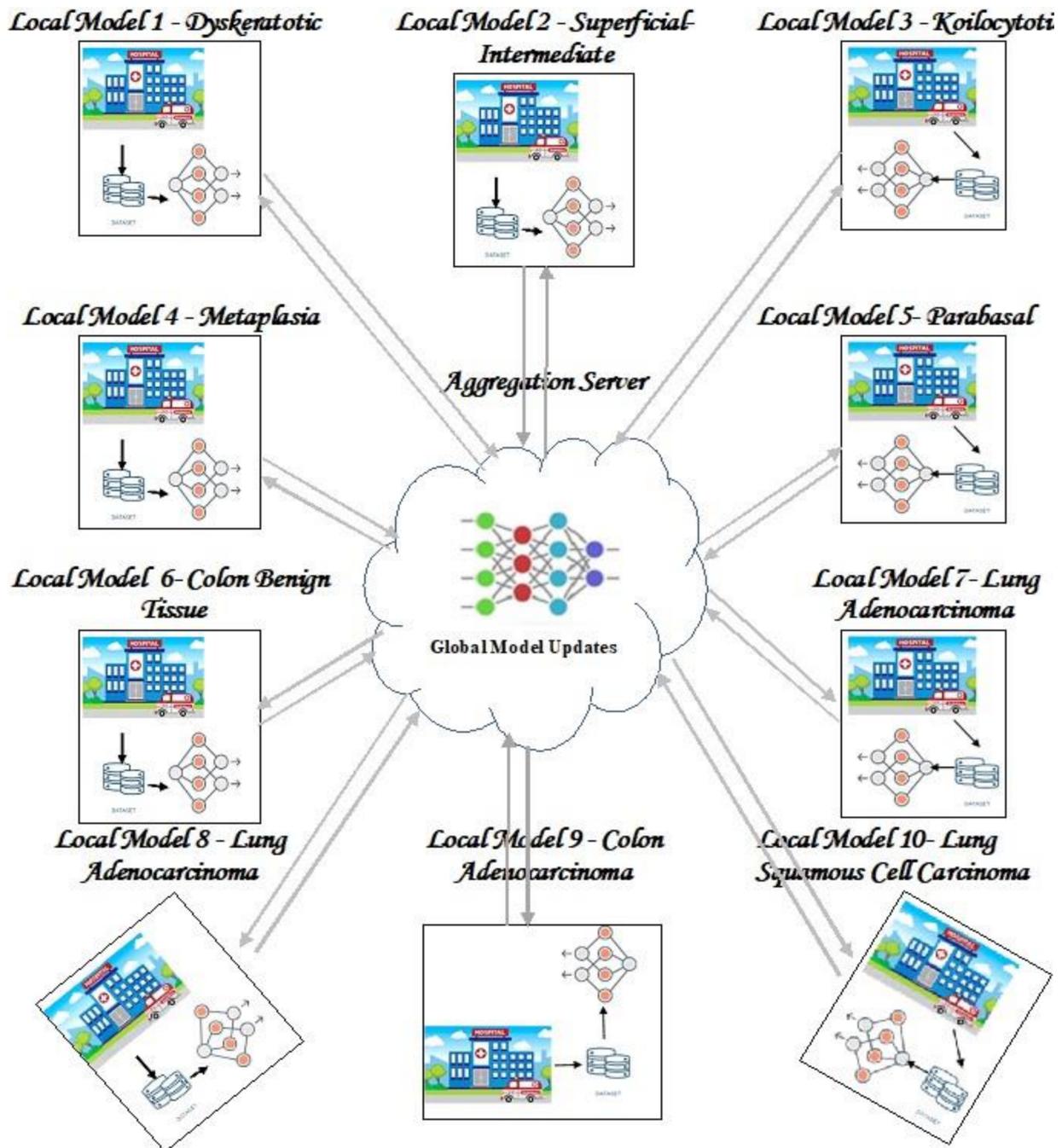


Figure 1. Proposed FL framework.

Figure 2 shows the training process in a local model. Apart from the FL algorithms used in this work, there are a few other FL algorithms proposed in different research attempts such as clustered FL [34,35], prototype-based FL [36,37], knowledge distillation-based FL [38,39], and pretrained FL [40,41]. In clustered FL [35], gradient descent has been used for optimizing the parameters of the local models. In [34], a multi-center aggregation wherein the local model’s parameters are aggregated at multiple local models having smaller intra-cluster distance was proposed. The FL server learns a lightweight generator to ensemble user information in a data-free manner as part of the data-free knowledge distillation approach [38,39]. This information is then disseminated to clients for additional

training. In our work, we used FedAvg and FedProx. The rationale for using these two algorithms include that FedAvg is one of the traditional vanilla FL algorithms formulated by Google and is simple to implement. The survey of these two FL algorithms demonstrates that FedProx is substantially better at handling non-IID data than FedAvg is, and it also converges to a higher accuracy in less training cycles [42]. Hence, we propose using these two FL algorithms.

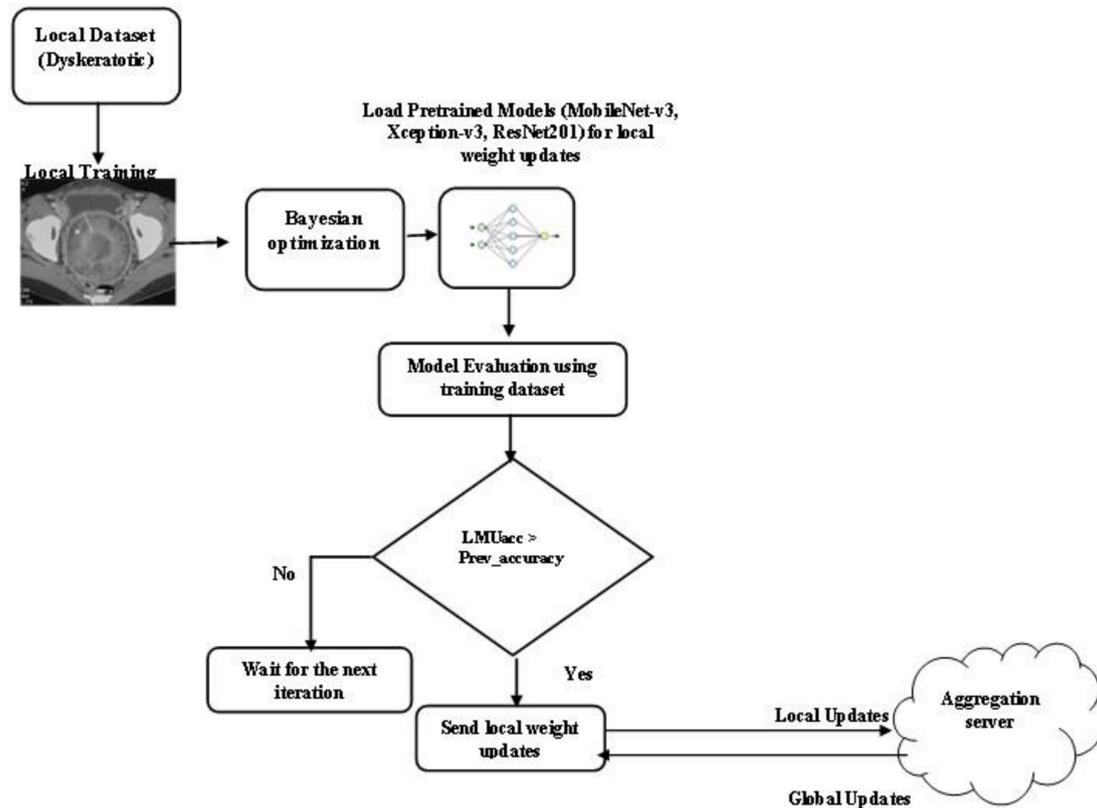


Figure 2. Training process in a client.

#### 4. Experimental Settings

The implementation of FL was carried out using cervical, lung, and colon cancer datasets containing MRI and CT, images. We built a Keras and TensorFlow-based FL training framework and placed it in a cloud-based, distributed system. The entire system was built with Tensorflow. Large models can be trained on the server’s GPU (a DELL EME 740 with enough GPU memory) while 10 CPU nodes train smaller CNNs on the client side.

##### 4.1. Dataset Description

CT/MRI images for the various body parts (tissues) were collected and compiled from Kaggle containing the categories of cervical, lung and colon cancer lesions. Since there were an unequal number of images in each class, we used image augmentation, a method for creating new images from the current training dataset artificially. Rotation, shifting, cropping, blurring, scaling, flipping both horizontal and vertical, and padding are examples of image augmentation methods to aid in minimizing the model’s over-fitting. A description of the dataset is presented in Table 2.

**Table 2.** Dataset description.

Cancer Types	Classes	Number of Training Images	Number of Testing Images
Cervical	Metaplasia	2000	500
	Dyskeratotic	2000	500
	Koilocytotic	2000	500
	Parabasal	2000	500
	Superficial–intermediate	2000	500
Lung	Lung benign tissue	2000	500
	Lung adenocarcinoma	2000	500
	Lung squamous cell carcinoma	2000	500
Colon	Colon adeno carcinoma	2000	500
	Colon benign tissue	2000	500

The training task was the classification of different imaging techniques such as MRI and CT scans having the traits of cervical, colon, and lung cancer. The research work generated a dataset with traits of different types of cancer. Then, non-identically distributed disjointed data collections were created from the generated dataset, i.e., IID. Each subset only included instances of a subset of the entire class set. Thus, we constructed a systematic partition of an initial data set such that the subsets were no longer IID and created the non-IID variants by splitting training images into 10 balanced partitions. The test images were used for a global test after each round. The dataset was divided into training and test sets with a ratio of 80:20. The training set was further split among the ten clients, each working on a type of cancer. This method of distribution provides a non-IID setting. It is referred to as feature shift non-IID when local clients each store images with distinct distributions from other clients. We considered local data to diverge from the feature space distribution, and we refer to this as a feature shift. In many real-world situations, this kind of non-IID data poses a serious difficulty, usually when the local devices are responsible for a heterogeneity in the feature distributions. All the clients had two non-similar labels, i.e., the distribution of the name of the labels varies between different nodes.

#### 4.2. Models

MobileNet-V3, Xception-V3, and ResNet201 were used to construct the classification models. These CNN variations were chosen because ResNet is constructed by piling residual blocks on top of one another and may have a larger number of layers, thereby learning all the parameters from early activations deeper into the network. The use of global average pooling in place of fully-connected layers makes ResNet significantly smaller. In Xception, depth-wise separable convolutions were used in place of the original Inception modules. For image classification, Xception surpasses Inception v3 by a small margin on the largest dataset having 17,000 classes, and by a much greater margin on the ImageNet dataset. Perhaps most significantly, it has the same amount of model parameters as Inception but is more computationally efficient. MobileNet, being a lightweight deep neural network, is both smaller in size and more accurate at classification.

#### 4.3. Hyperparameter Tuning

We applied Bayesian optimization to optimize hyperparameters [43,44]. It uses Bayes' theorem to find the minimum or maximum of an objective function. This study minimizes the loss function. Bayesian optimization uses the Gaussian process (GP) to determine good hyperparameter values. GP generates data from the hyperparameter search space and formulates a hypothesis based on known parameters. The Bayesian strategy may take longer to pick hyperparameters, but it takes less time to evaluate the objective function, reducing the computational cost. The optimizer, learning rate, activation function, numbers

of hidden layers, batch sizes, epochs, number of neurons, and communication rounds were fine-tuned. Table 3 outlines our work's hyperparameters and search spaces.

**Table 3.** Hyper-parameters and their search spaces.

Parameter	Search Space
Optimizer	Adam, RMSProp, SGD
Learning rate	$1 \times 10^{-3}$ , $1 \times 10^{-4}$ , $1 \times 10^{-5}$ , $1 \times 10^{-6}$
Activation function	Relu, Elu, and Tanh
Number of neurons in customized layers	64, 128, 56, 512, 1024
Number of hidden layers	2, 3, 4, 5
Batch Size	32, 64, 128
Epochs	10, 20, 30, 40
Communication rounds	100, 150, 200, 250

## 5. Experimental Results

### 5.1. Performance Evaluation

Our evaluations include the comparison between the performance of FedAvg and FedProx algorithms [2,8,32] and the hyperparameter optimization approaches mentioned in Section 4. To train the proposed FL framework, at first, we split the training datasets into 10 parts, each with images of a specific type of cervical/lung/colon cancer and loaded them into ten clients. The clients use MobileNetV3, XceptionV3 and ResNet201 for the classification of images. We ran the FL models using the values of the hyperparameters shown in Table 3. The optimal values of these hyperparameters were obtained using Bayesian optimization and are shown in Table 4.

**Table 4.** Hyperparameters and their optimal values.

Parameters	Optimal Values
Optimizer	SGD
Learning rate (client and server)	$1 \times 10^{-3}$ (client) and $1 \times 10^{-4}$ (server)
Activation function (client)	Relu
Number of neurons in customized layers (client)	128
Number of hidden layers (client)	4
Batch Size (client)	128
Epochs (client)	10
Communication rounds (between client and server)	150

We now present empirical results for the FedAvg and FedProx framework. We ran the global model using the test data and present the results. Figure 3 shows the accuracy obtained for the FedAvg FL algorithm for a varied number of communication rounds. The values of the loss function for these runs are also depicted in Figure 4. The confusion matrix obtained for FedAvg is shown in Figure 5.

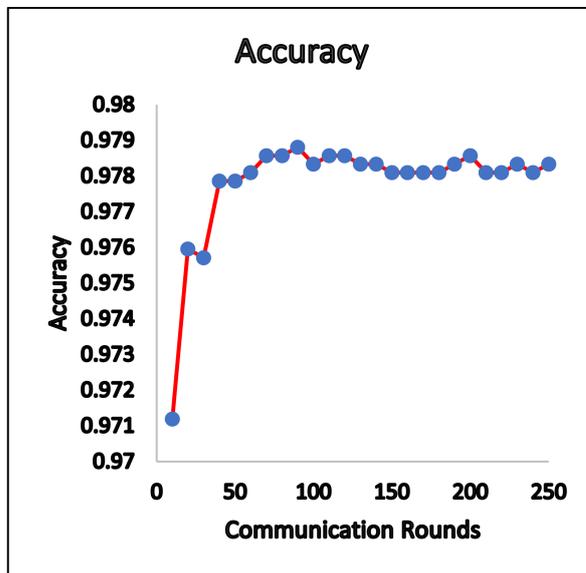
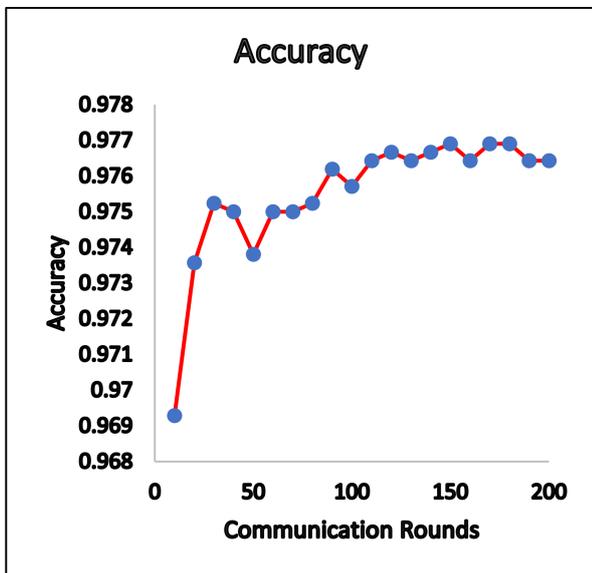
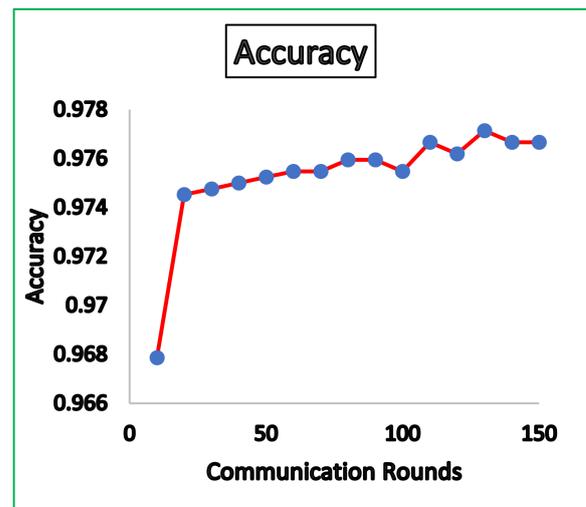
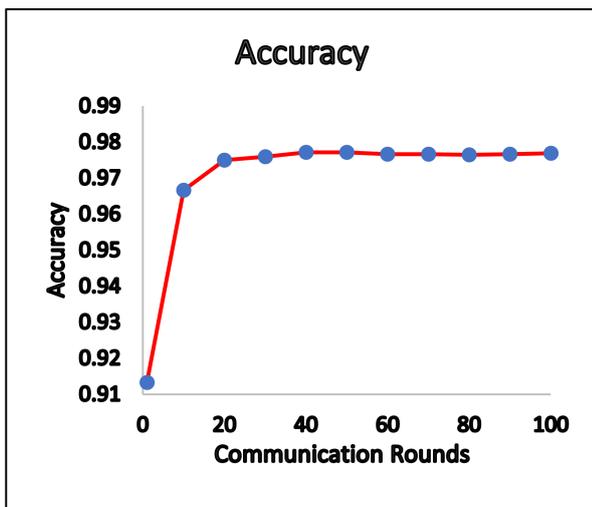


Figure 3. FedAvg—accuracy.

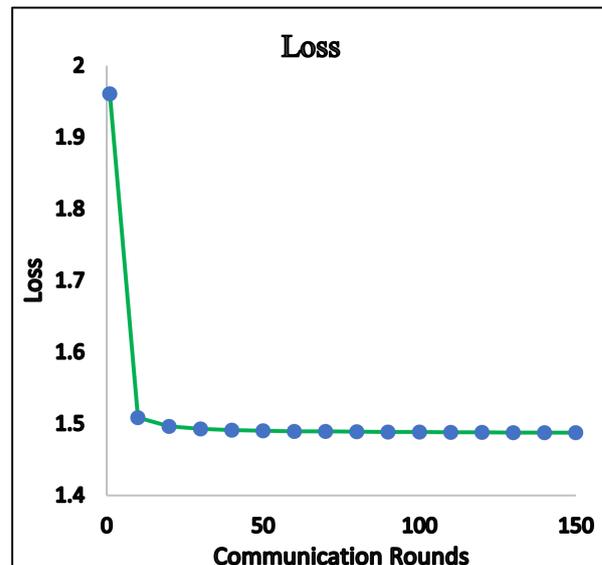
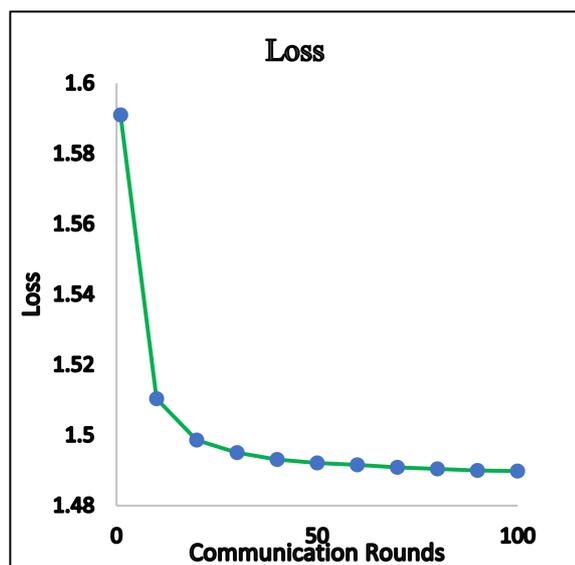


Figure 4. Cont.

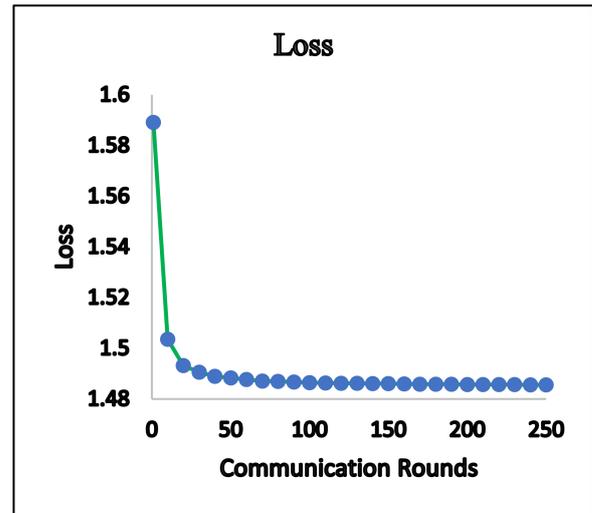
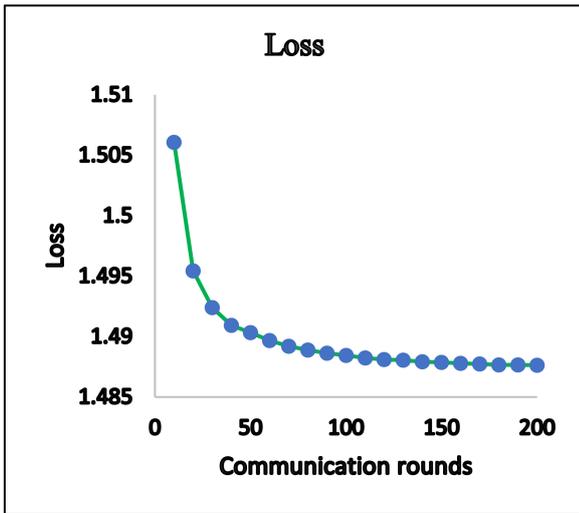
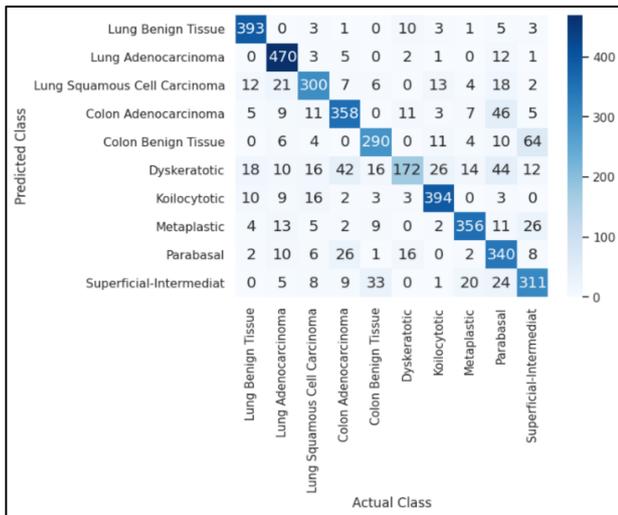
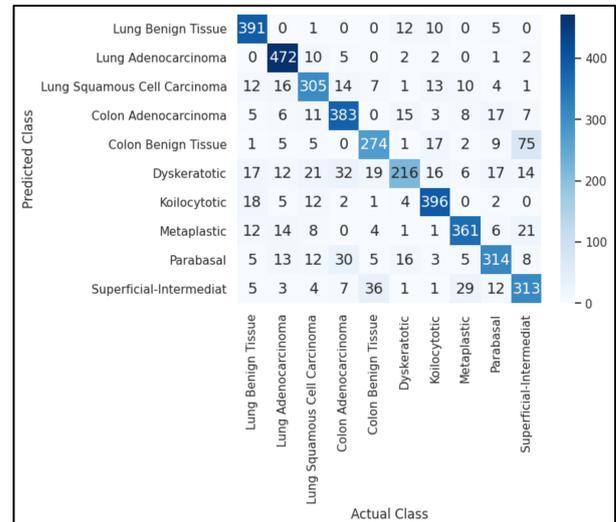


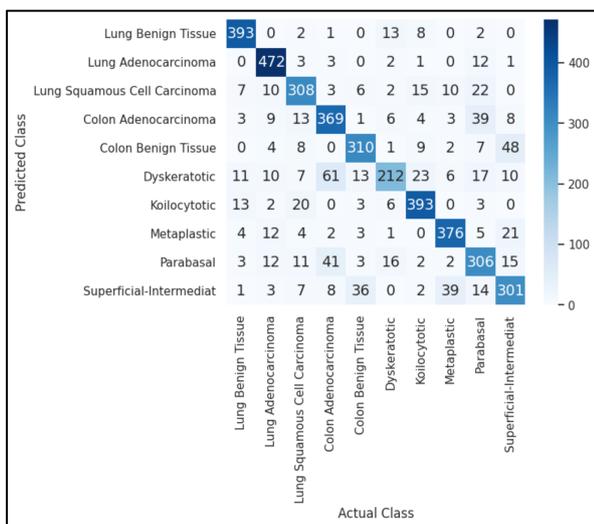
Figure 4. FedAvg—loss.



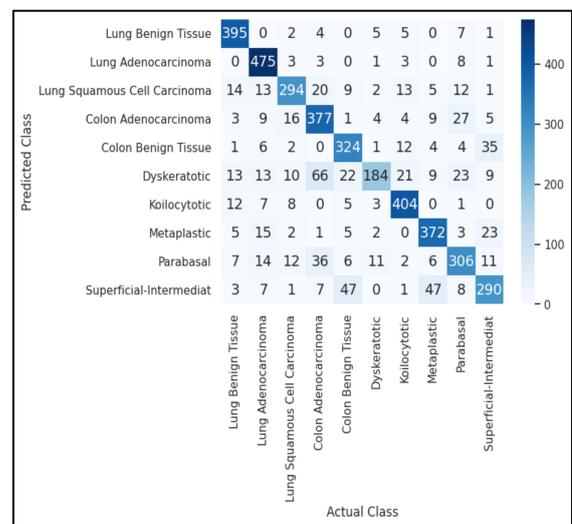
(a) Communication rounds—250



(b) Communication rounds—200



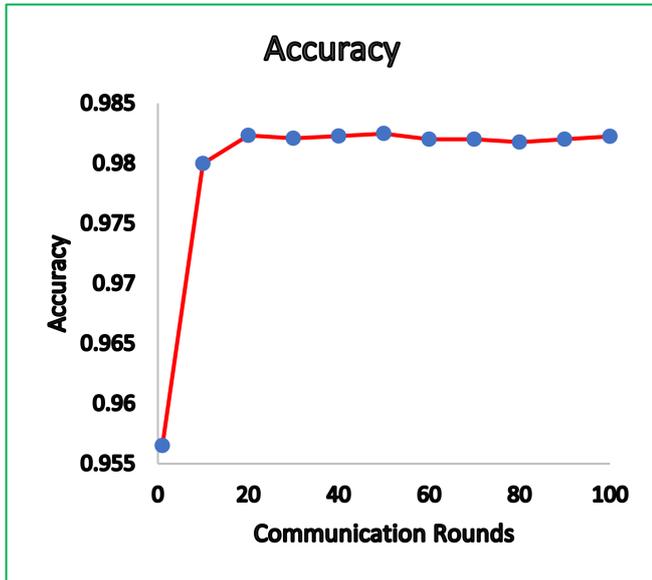
(c) Communication rounds—150



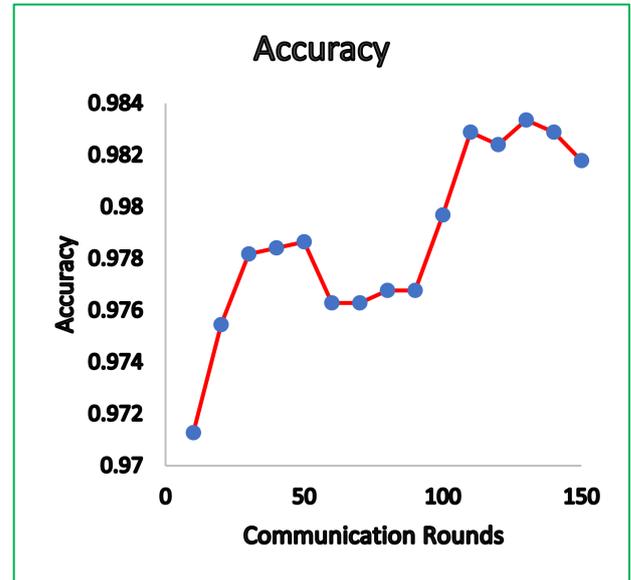
(d) Communication rounds—100

Figure 5. FedAvg—confusion matrix.

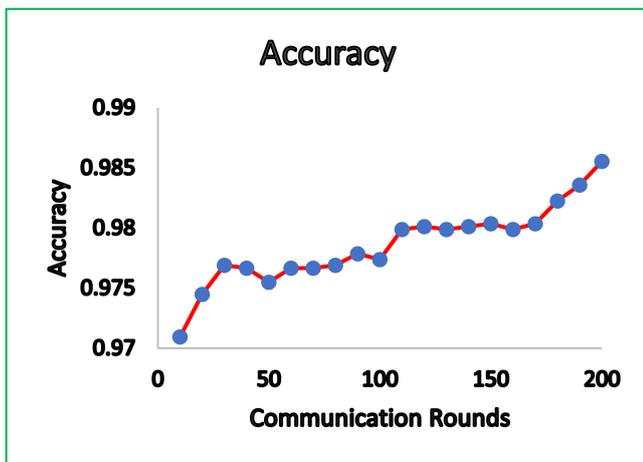
We also ran the FedProx FL algorithm and obtained the classification results. The accuracy and loss function values for different communication rounds are presented in Figures 6 and 7. The confusion matrix obtained for FedProx is shown in Figure 8.



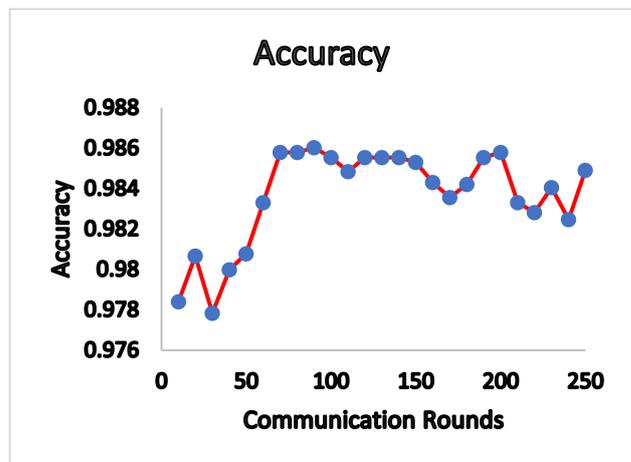
(a) Communication rounds—100



(b) Communication rounds—150



(c) Communication rounds—200



(d) Communication rounds—250

Figure 6. FedProx—accuracy.

In order to visualize the difference between the performance of FedAvg and FedProx, we projected and showed the accuracy and loss of both algorithms in Figures 9 and 10.

After the models were developed, we evaluated the performance of the proposed FL models in terms of accuracy, precision, recall, and F1 Score [43,44]. The values of these metrics were determined using indices such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The results of the evaluation are presented in Table 5.

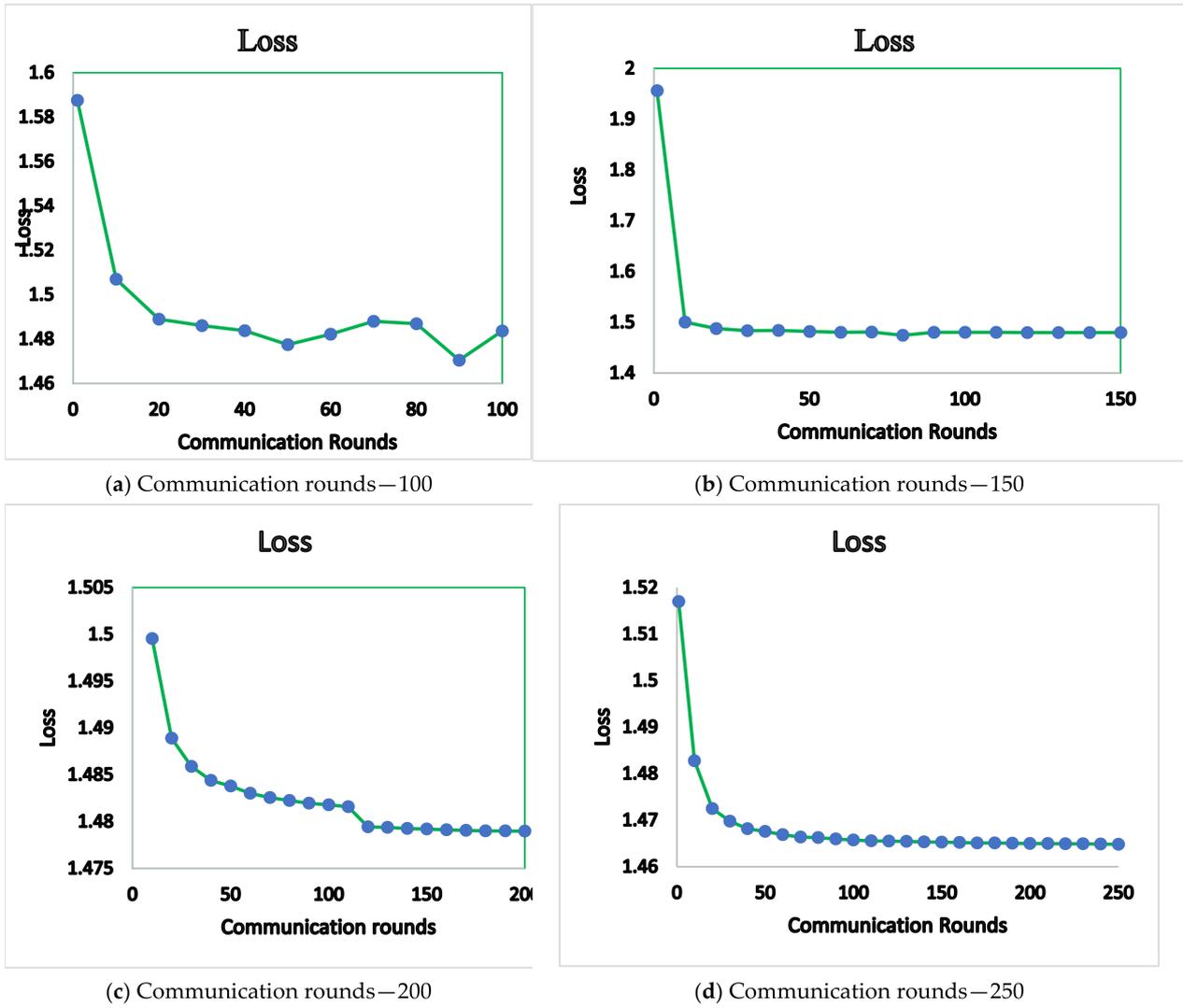


Figure 7. FedProx—loss.

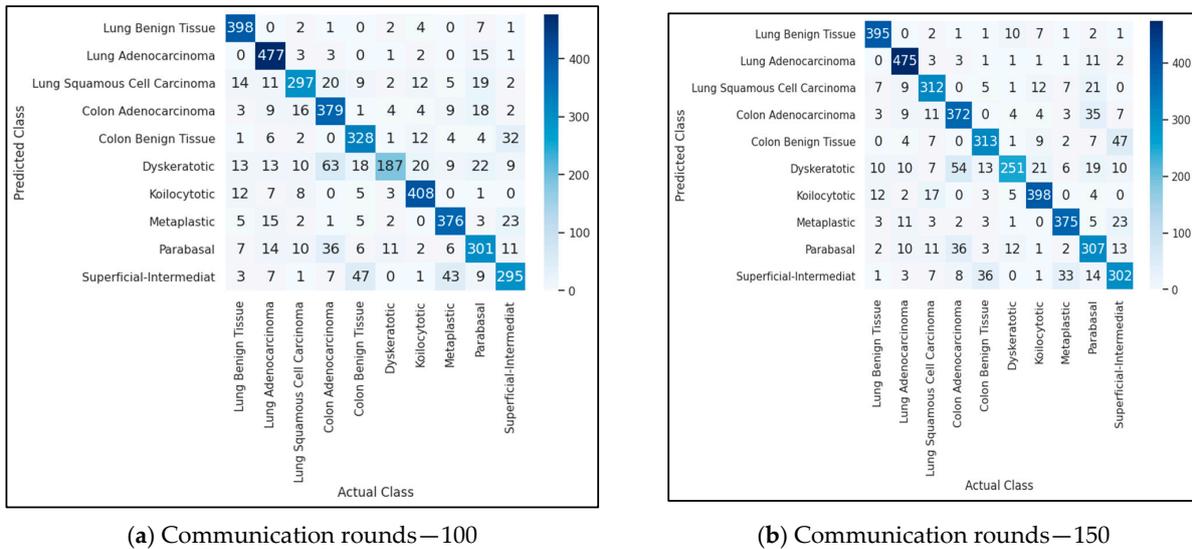
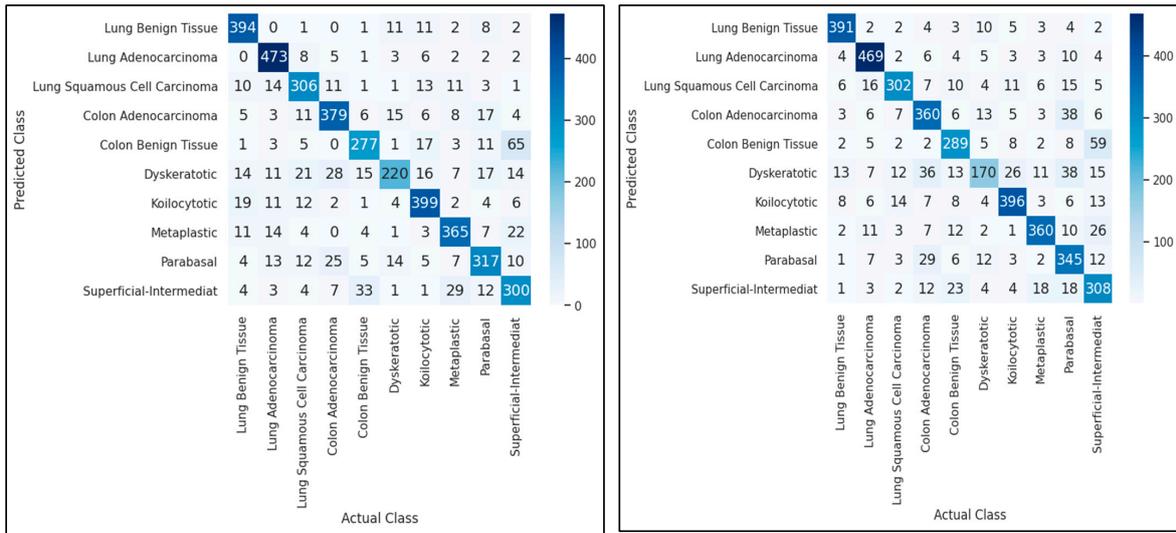


Figure 8. Cont.



(c) Communication rounds—200

(d) Communication rounds—250

Figure 8. FedProx—confusion matrix.

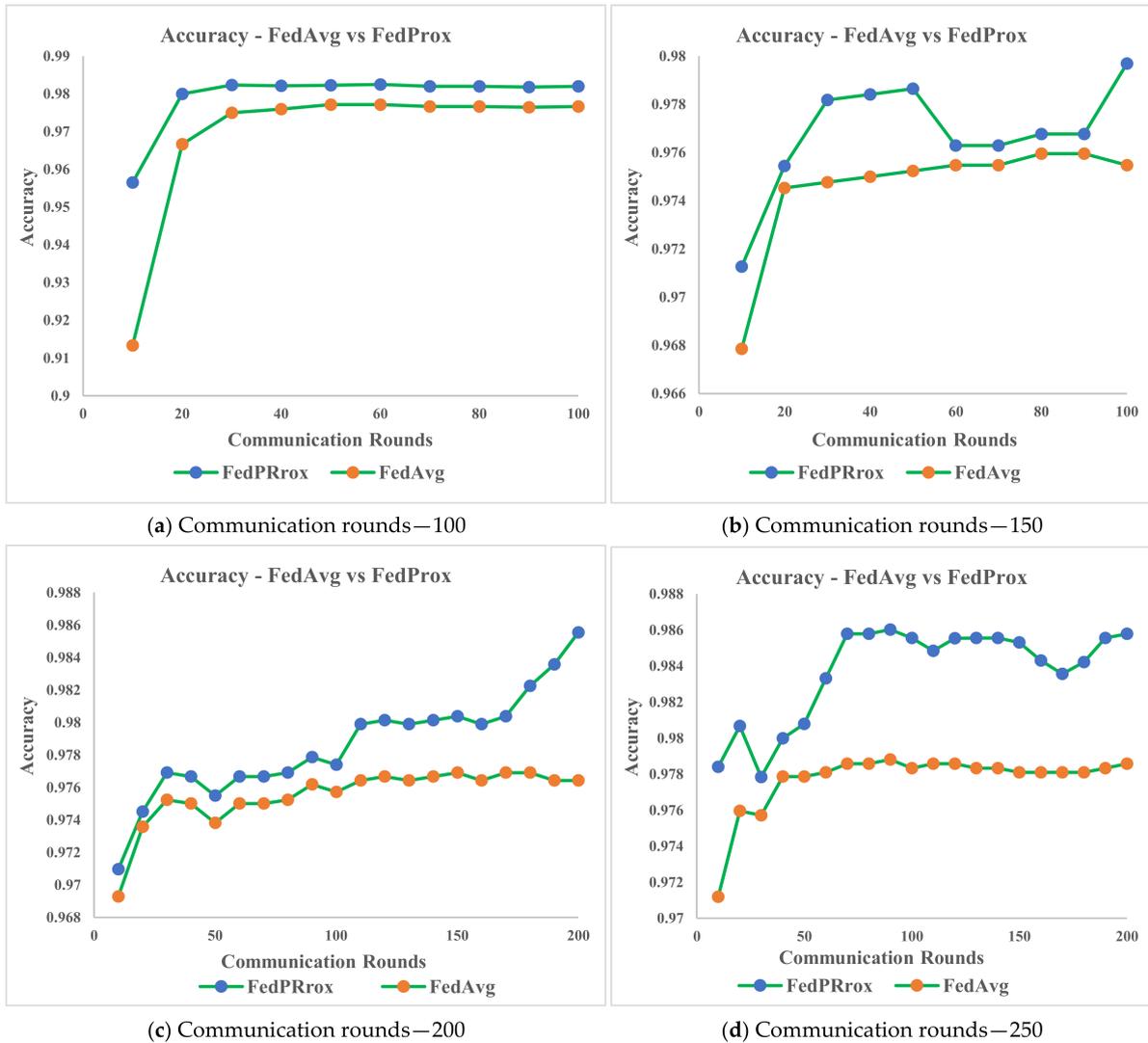


Figure 9. FedAvg vs. FedProx—accuracy.

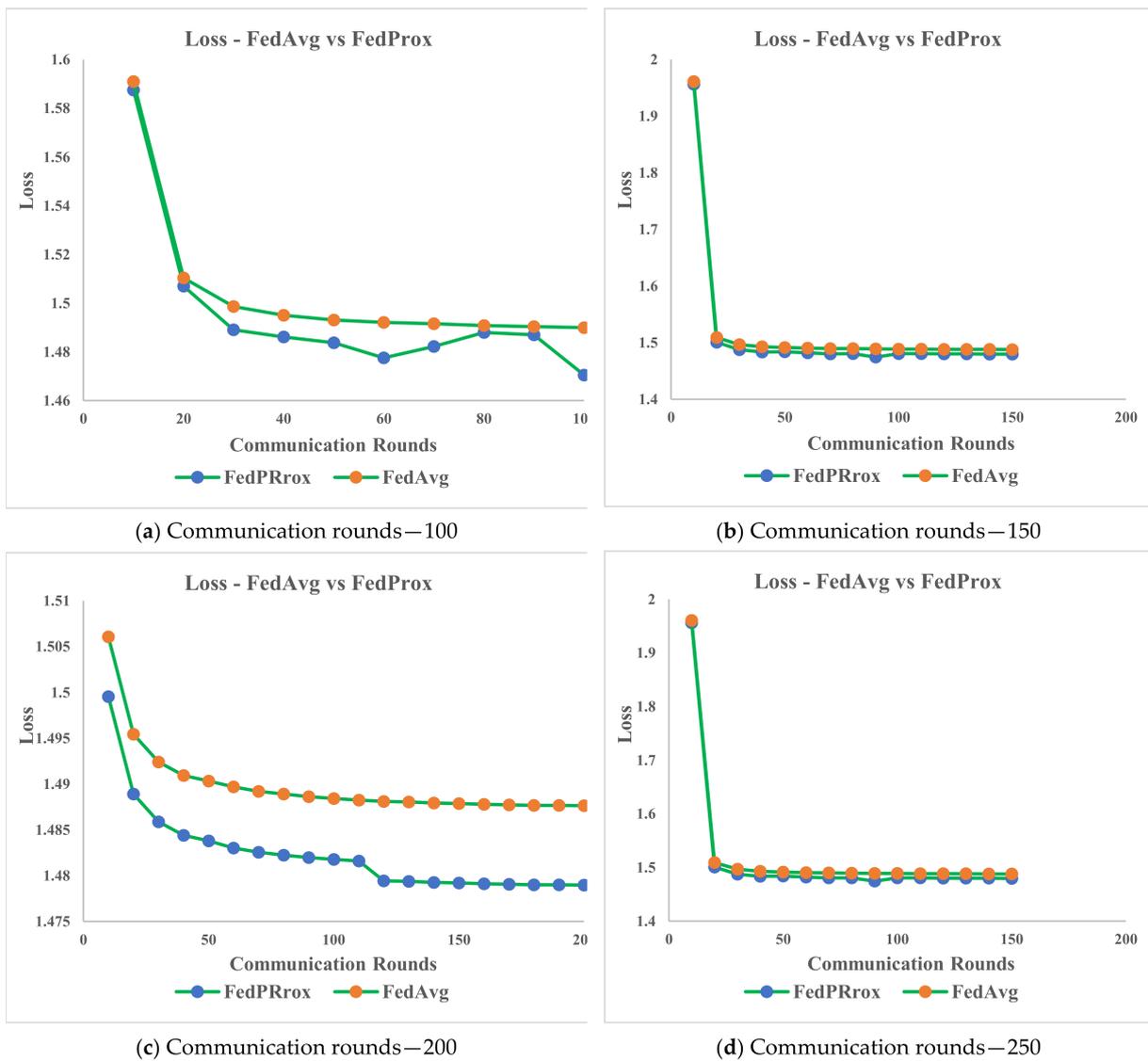


Figure 10. FedAVg vs. FedProx—loss.

Table 5. FedAvg vs. FedProx.

Communication Rounds	FedAvg			FedProx		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
250	93.79	88.51	91.08	91.78	90.72	91.25
	95.14	84.99	89.78	91.96	88.16	90.02
	78.33	80.65	79.47	79.06	86.53	82.63
	78.68	79.2	78.94	80.54	76.6	78.52
	74.55	81.01	77.64	75.65	77.27	76.46
	46.49	80.37	58.9	49.85	74.24	59.65
	89.55	86.78	88.14	85.16	85.71	85.44
	83.18	87.25	85.17	82.95	87.59	85.21
	82.73	66.28	73.59	82.14	70.12	75.66
	75.67	71.99	73.78	78.37	68.44	73.07

Table 5. Cont.

Communication Rounds	FedAvg			FedProx		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
200	93.32	83.91	88.36	91.63	85.28	88.34
	95.55	86.45	90.77	94.22	86.79	90.35
	79.63	78.41	79.02	82.48	79.69	81.06
	84.18	80.97	82.54	83.48	82.93	83.21
	70.44	79.19	74.56	72.32	80.52	76.2
	58.38	80.3	67.61	60.61	81.18	69.4
	90	85.71	87.8	86.74	83.65	85.17
	84.35	85.75	85.04	84.69	83.72	84.2
	76.4	81.14	78.7	76.94	79.65	78.27
	76.16	70.98	73.47	76.14	70.42	73.17
150	93.79	90.34	92.04	94.05	91.22	92.61
	95.55	88.39	91.83	95.38	89.12	92.14
	80.42	80.42	80.42	83.42	82.11	82.76
	81.1	75.61	78.26	83.04	78.15	80.52
	79.69	82.67	81.15	80.26	82.8	81.51
	57.3	81.85	67.41	62.59	87.76	73.07
	89.32	86	87.63	90.25	87.67	88.94
	87.85	85.84	86.84	88.03	87.21	87.62
	74.45	71.66	73.03	77.33	72.24	74.7
	73.24	74.5	73.87	74.57	74.57	74.57
100	94.27	87.2	90.6	95.9	87.28	91.39
	96.15	84.97	90.22	95.02	85.33	89.92
	76.76	84	80.22	75.96	84.62	80.05
	82.86	73.35	77.81	85.17	74.31	79.37
	83.29	77.33	80.2	84.1	78.28	81.09
	49.73	86.38	63.12	51.37	87.79	64.82
	91.82	86.88	89.28	91.89	87.74	89.77
	86.92	82.3	84.55	87.04	83.19	85.07
	74.45	76.69	75.56	74.5	75.44	74.97
	70.56	77.13	73.7	71.43	78.46	74.78

In Table 6, we present the testing accuracy of FedAvg and FedProx on our datasets associated with the experiments discussed in Section 4.

Table 6. Performance comparison of FedAvg and FedProx on test data—accuracy.

Communication Rounds	FedAvg	FedProx
100	81.45%	82.05%
150	81.91%	83.313%
200	81.548%	81.667%
250	80.57%	80.73%

## 5.2. Findings and Discussion

In this section, we intend to provide the findings of the above experimental results. We noticed that the performance of FedProx was greatly affected by the value we chose for  $\mu$ . Using  $\mu = 0.1, 0.2, 0.3, 0.01$ , and  $1$ , we ran trials. It was deduced from these experiments that while a large value of  $\mu$  can hasten the convergence, it can also miss global minima. However, convergence time increased for smaller values. Experiments showed that a value of  $0.2$  for  $\mu$  provides optimal performance. We also determined the training time for FedAvg and FedProx algorithms, given that the global update was broadcast to all clients simultaneously. The training period in this experiment was found to be marginally longer than the FedAvg employed in [8], where a random selection of clients was selected by the server to receive global updates to parameters. While this is true in FedAvg and helps to reduce the training time, we discover that it may not carry out global updates as smoothly as we would like. Furthermore, in our proposed FL architecture, only the clients who obtained better accuracy for the newly calculated local weight parameters communicated the global updates to the server for the subsequent rounds. Then, the necessary changes were made to the global model.

Furthermore, we also compared the performance of the FL algorithms with that of traditional deep learning models. For this trial, we ran the datasets on ten clients each with a dataset of a specific type of cancer and found that the models on clients performed well for the dataset on which they have been trained. But these models did not exhibit good performance for the unseen type of cancer. Then, when run using FL algorithms we obtained an improved performance as illustrated in Section 5. While analyzing the results for different communication rounds, we found that when the number of communication rounds increased, there was a degradation in the performance of the FL model. As we increased the number of communication rounds, we understood that the updates in local parameters had reduced the accuracy of the local model, which in turn impacted the performance of the global model. We also found that FedProx outperformed FedAvg in terms of all the performance metrics. Since, in FedProx, the local models use their previous weight parameters along with global parameters to update the parameters for the current iteration, it showed improved performance. This was achieved with the help of the proximal term ( $\mu$ ). This term solves the problem of statistical heterogeneity by limiting local updates to be closer to the global model. Even though it is still hard to make good models with small datasets at each site [45], our model did better with proper training and tuning of the hyperparameters.

Additionally, we also tested the performance of the proposed models on the benchmark dataset MNIST. For this test, we used the IID version of the MNIST dataset (each client has equal number of images and images with all class labels). During this evaluation, we found that, for the MNIST dataset, our models had an accuracy of  $87.16\%$ . Since this dataset offers a public benchmark extracted from over a million human-labelled and curated examples, we believe that the results are appreciable.

To gain a better understanding of the difficulties inherent in implementing the FL, we also looked at errors brought on by the algorithms. True positives, false positives, true negatives, and false negatives are all categories for the outcomes of the classification models on images. From the confusion matrix shown in Figure 5, we observed that true positive for the class colon adenocarcinoma was 383. This means that out of 500 colon adenocarcinoma images, 383 images were correctly classified, and the remaining 127 images were misclassified. However, there was a small increase in the proportion of correctly identified images using the FedProx algorithm. While analyzing the reasons for misclassification, we found that the similarity among benign and malignant cancer lesions was very high, and this led to misclassification. Another reason could be the feature map learned by the algorithm. Since a few benign images resembled malignant images, the feature maps generated for both types were indistinguishable. Thus, making use of the incorrectly classified images can assist in increasing classification precision.

Intense interest has been generated by FL, a developing collaborative AI technique, to achieve scalable and privacy-enhancing smart healthcare networks and applications. In this study, we applied the FedAvg and FedProx FL algorithms to three heterogeneous cancer types and found that FL models outperform conventional deep learning models in terms of prediction accuracy. Out of these two algorithms, the FedProx FL algorithm outperformed the FedAvg algorithm. We tested the FL algorithms with varying numbers of communication rounds and understood that performance did not improve after a certain number of rounds. In addition, because of the distributed nature of FL algorithms, the training period is significantly longer.

Although our work does not address the privacy concern, we believe existing methods such as differential privacy and multi-party computation can be used to extend our work. We also plan to extend our work by considering more heterogeneous datasets. However, the application of FL is still in its infancy and will quickly mature in the coming years to provide intelligent and privacy-enhanced services.

**Author Contributions:** Data curation: M.S. and K.S.; Formal analysis: S.V.E. and M.S.; Investigation: V.R. and K.S.; Methodology: P.S.N. and M.S.; Project administration: S.V.E. and V.R.; Validation: K.S.; Visualization: S.V.E.; Writing—original draft preparation: M.S. and V.R.; Writing—review and editing: S.V.E. and P.S.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research did not receive funding from any sources.

**Data Availability Statement:** The dataset used in this research will be made available to the public once the work is published. However, the data presented in this study are available on request from the corresponding author.

**Acknowledgments:** We would also like to render our acknowledgment to Obuli Sai Naren, Prem Kumar, Rankish, and UG CSE students for their support in developing the proposed models.

**Conflicts of Interest:** The authors declare no potential conflicts of interest.

## References

1. Sheller, M.J.; Edwards, B.; Reina, G.A.; Martin, J.; Pati, S.; Kotrotsou, A.; Milchenko, M.; Xu, W.; Marcus, D.; Colen, R.R. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Sci. Rep.* **2020**, *10*, 12598. [[CrossRef](#)] [[PubMed](#)]
2. Khan, L.U.; Saad, W.; Han, Z.; Hossain, E.; Hong, C.S. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1759–1799. [[CrossRef](#)]
3. Zhu, Y.; Shen, B.; Pei, X.; Liu, H.; Li, G. CT, MRI, and PET imaging features in cervical cancer staging and lymph node metastasis. *Am. J. Transl. Res.* **2021**, *13*, 10536. [[PubMed](#)]
4. Nguyen, D.C.; Pham, Q.-V.; Pathirana, P.N.; Ding, M.; Seneviratne, A.; Lin, Z.; Dobre, O.; Hwang, W.-J. Federated learning for smart healthcare: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 1–37. [[CrossRef](#)]
5. Shaheen, M.; Farooq, M.S.; Umer, T.; Kim, B.-S. Applications of federated learning; Taxonomy, challenges, and research trends. *Electronics* **2022**, *11*, 670. [[CrossRef](#)]
6. He, C.; Annavaram, M.; Avestimehr, S. Group knowledge transfer: Federated learning of large cnns at the edge. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 14068–14080.
7. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Poor, H.V. Federated learning for internet of things: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1622–1658. [[CrossRef](#)]
8. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
9. Jeong, E.; Oh, S.; Kim, H.; Park, J.; Bennis, M.; Kim, S.-L. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv* **2018**, arXiv:1811.11479.
10. Shen, T.; Zhang, J.; Jia, X.; Zhang, F.; Huang, G.; Zhou, P.; Kuang, K.; Wu, F.; Wu, C. Federated mutual learning. *arXiv* **2020**, arXiv:2006.16765.
11. Liu, L.; Zhang, J.; Song, S.; Letaief, K.B. Client-edge-cloud hierarchical federated learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
12. Li, T.; Sanjabi, M.; Beirami, A.; Smith, V. Fair resource allocation in federated learning. *arXiv* **2019**, arXiv:1905.10497.
13. Mohammadi, R.; Shokatian, I.; Salehi, M.; Arabi, H.; Shiri, I.; Zaidi, H. Deep learning-based auto-segmentation of organs at risk in high-dose rate brachytherapy of cervical cancer. *Radiother. Oncol.* **2021**, *159*, 231–240. [[CrossRef](#)]
14. Zhang, T.; Gao, L.; He, C.; Zhang, M.; Krishnamachari, B.; Avestimehr, A.S. Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities. *IEEE Internet Things Mag.* **2022**, *5*, 24–29. [[CrossRef](#)]

15. Zhang, L.; Luo, Y.; Bai, Y.; Du, B.; Duan, L.-Y. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4420–4428.
16. Holly, S.; Hiessl, T.; Lakani, S.R.; Schall, D.; Heitzinger, C.; Kemnitz, J. Evaluation of hyperparameter-optimization approaches in an industrial federated learning system. In *Data Science–Analytics and Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 6–13.
17. Guo, P.; Yang, D.; Hatamizadeh, A.; Xu, A.; Xu, Z.; Li, W.; Zhao, C.; Xu, D.; Harmon, S.; Turkbey, E. Auto-FedRL: Federated Hyperparameter Optimization for Multi-institutional Medical Image Segmentation. *arXiv* **2022**, arXiv:2203.06338.
18. Zhu, H.; Xu, J.; Liu, S.; Jin, Y. Federated learning on non-IID data: A survey. *Neurocomputing* **2021**, *465*, 371–390. [[CrossRef](#)]
19. Kumar, R.; Wang, W.; Yuan, C.; Kumar, J.; Qing, H.; Yang, T.; Khan, A.A. Blockchain based privacy-preserved federated learning for medical images: A case study of COVID-19 CT scans. *arXiv* **2021**, arXiv:2104.10903.
20. Dou, Q.; So, T.Y.; Jiang, M.; Liu, Q.; Vardhanabhuti, V.; Kaissis, G.; Li, Z.; Si, W.; Lee, H.H.; Yu, K. Federated deep learning for detecting COVID-19 lung abnormalities in CT: A privacy-preserving multinational validation study. *NPJ Digit. Med.* **2021**, *4*, 1–11. [[CrossRef](#)]
21. Kumar, R.; Khan, A.A.; Kumar, J.; Golilarz, N.A.; Zhang, S.; Ting, Y.; Zheng, C.; Wang, W. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. *IEEE Sens. J.* **2021**, *21*, 16301–16314. [[CrossRef](#)]
22. Yang, Q.; Zhang, J.; Hao, W.; Spell, G.P.; Carin, L. Flop: Federated learning on medical datasets using partial networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 3845–3853.
23. Salam, M.A.; Taha, S.; Ramadan, M. COVID-19 detection using federated machine learning. *PLoS ONE* **2021**, *16*, e0252573.
24. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Zomaya, A.Y. Federated learning for COVID-19 detection with generative adversarial networks in edge cloud computing. *IEEE Internet Things J.* **2022**, *9*, 10257–10271. [[CrossRef](#)]
25. Beguier, C.; Du Terrail, J.O.; Meah, I.; Andreux, M.; Tramel, E.W. Differentially private federated learning for cancer prediction. *arXiv* **2021**, arXiv:2101.02997.
26. Choudhury, O.; Gkoulalas-Divanis, A.; Salonidis, T.; Sylla, I.; Park, Y.; Hsu, G.; Das, A. Differential privacy-enabled federated learning for sensitive health data. *arXiv* **2019**, arXiv:1910.02578.
27. Hansen, C.R.; Price, G.; Field, M.; Sarup, N.; Zukauskaitė, R.; Johansen, J.; Eriksen, J.G.; Aly, F.; McPartlin, A.; Holloway, L. Larynx cancer survival model developed through open-source federated learning. *Radiother. Oncol.* **2022**, *176*, 179–186. [[CrossRef](#)] [[PubMed](#)]
28. Brisimi, T.S.; Chen, R.; Mela, T.; Olshevsky, A.; Paschalidis, I.C.; Shi, W. Federated learning of predictive models from federated electronic health records. *Int. J. Med. Inform.* **2018**, *112*, 59–67. [[CrossRef](#)]
29. Kumar, Y.; Singla, R. Federated learning systems for healthcare: Perspective and recent progress. In *Federated Learning Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 141–156.
30. Long, G.; Shen, T.; Tan, Y.; Gerrard, L.; Clarke, A.; Jiang, J. Federated learning for privacy-preserving open innovation future on digital health. In *Humanity Driven AI*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 113–133.
31. Antunes, R.S.; Da Costa, C.A.; Küderle, A.; Yari, I.A.; Eskofier, B. Federated Learning for Healthcare: Systematic Review and Architecture Proposal. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–23. [[CrossRef](#)]
32. Rieke, N.; Hancox, J.; Li, W.; Milletari, F.; Roth, H.R.; Albarqouni, S.; Bakas, S.; Galtier, M.N.; Landman, B.A.; Maier-Hein, K. The future of digital health with federated learning. *NPJ Digit. Med.* **2020**, *3*, 119. [[CrossRef](#)]
33. Yuan, X.-T.; Li, P. On Convergence of FedProx: Local Dissimilarity Invariant Bounds, Non-smoothness and Beyond. *arXiv* **2022**, arXiv:2206.05187.
34. Long, G.; Xie, M.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J. Multi-center federated learning: Clients clustering for better personalization. *World Wide Web* **2022**, 1–20. [[CrossRef](#)]
35. Ghosh, A.; Chung, J.; Yin, D.; Ramchandran, K. An efficient framework for clustered federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19586–19597. [[CrossRef](#)]
36. Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In Proceedings of the AAAI Conference on Artificial Intelligence, Pomona, CA, USA, 24–28 October 2022; p. 3.
37. Chen, Z.; Yang, C.; Zhu, M.; Peng, Z.; Yuan, Y. Personalized retrogress-resilient federated learning towards imbalanced medical data. *IEEE Trans. Med. Imaging* **2022**, *41*, 3663–3674. [[CrossRef](#)]
38. Zhu, Z.; Hong, J.; Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. *Proc. Mach. Learn. Res.* **2021**, *139*, 12878–12889.
39. Zhang, L.; Shen, L.; Ding, L.; Tao, D.; Duan, L.-Y. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10174–10183.
40. Tan, Y.; Long, G.; Ma, J.; Liu, L.; Zhou, T.; Jiang, J. Federated learning from pre-trained models: A contrastive learning approach. *arXiv* **2022**, arXiv:2209.10083.
41. Chen, H.-Y.; Tu, C.-H.; Li, Z.; Shen, H.-W.; Chao, W.-L. On pre-training for federated learning. *arXiv* **2022**, arXiv:2206.11488.
42. Aquilante, J.; Jicha, A. Heterogeneity Aware Federated Learning. Bachelor’s Thesis, Worcester Polytechnic Institute, Worcester, MA, USA, 2021.

43. Subramanian, M.; Shanmugavadivel, K.; Nandhini, P. On fine-tuning deep learning models using transfer learning and hyperparameters optimization for disease identification in maize leaves. *Neural Comput. Appl.* **2022**, *34*, 13951–13968. [[CrossRef](#)]
44. Subramanian, M.; Prasad, L.V.N.; Janakiramaiah, B.; Babu, A.M.; Sathishkumar, V.E. Hyperparameter optimization for transfer learning of VGG16 for disease identification in corn leaves using Bayesian optimization. *Big Data* **2022**, *10*, 215–229. [[CrossRef](#)]
45. Ng, D.; Lan, X.; Yao, M.M.-S.; Chan, W.P.; Feng, M. Federated learning: A collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets. *Quant. Imaging Med. Surg.* **2021**, *11*, 852. [[CrossRef](#)]