Privacy Focused Classification of Prostate Cancer Using Federated Learning

by

Syeda Umme Salma
18101350
MD. Sadman Sakib
18101089
Mohammed Moinul Morshed Alvee
18101077
Nahiyan Yasaar
21141018

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Syeda Umme Salma	MD. Sadman Sakib
Syeda Umme Salma 18101350	MD. Sadman Sakib 18101089
	Nahiyan Yasaar
Mohammed Moinul Morshed Alvee 18101077	Nahiyan Yasaar 21141018

Approval

The thesis/project titled "Privacy Focused Classification of Prostate Cancer Using Federated Learning" submitted by

1. Syeda Umme Salma (18101350)
2. MD. Sadman Sakib (18101089)
3. Mohammed Moinul Morshed Alvee (18101077)
4. Nahiyan Yasaar (21141018)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 20, 2022.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez

Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University,
ou=CSE, email=zavid.parvez@bracu.ac.bd,
c=US
Date: 2022.01.20 19:59:58 +11'00'

Dr. Mohammad Zavid Parvez
Former Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Tanzim

Md. Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The prostate gland is a small gland located in the lower abdomen of a man. Prostate cancer occurs when a tumor, or abnormal, malignant growth of cells, forms in the prostate. Prostate cancer is a slow-growing cancer that often goes undetected until it has progressed to an advanced stage. The majority of men with prostate cancer are unaware of having it, and many of them die of other causes before they even get diagnosed with it. However, prostate cancer becomes hazardous when it grows rapidly or spreads outside of the prostate. With early detection and personalized care, the prostate cancer survival rate is significantly increased. Deep learning can play a significant role regarding this, as the field of medical imaging has shown that identification based on computer-aided diagnosis helps radiologists make more precise diagnoses while still reducing diagnostic time and costs. However, the data concerning prostate cancer can be quite difficult to collect and it is used in a restricted manner due to the unwillingness of the patients to share and the hospital's confidentiality about their patients' records. The aim of our research was to address these challenges and it led us to develop such a system where prostate cancer can be classified, maintaining confidentiality of the data using a decentralized method called federated learning, different from how it can be done with current approaches. In this research, we have classified prostate cancer using simple CNN, Xception and VGG19 models in both traditional and federated learning approaches for comparative analysis. In fact, VGG19 outperformed the other two models in both approaches, with centralized classification accuracy being 95.51% and decentralized classification accuracy being 83.76%. Most importantly, through our system, the instance of our server-side model is distributed to different clients so that the clients can independently train their model using their local dataset in their own environment. Eventually, the updated weights of those trained models return back to the server to be aggregated from all the contemporary clients to finally train our server-side model without even accessing confidential medical data in order to ensure privacy focused classification.

Keywords: Federated Learning; Prostate Cancer; Secure Deep Learning; Privacy; Distributed Learning; Medical Imaging

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our supervisor Mohammad Zavid Parvez and co-advisor Md. Tanzim Reza sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ADR	Adverse Drug Reaction
AFL	Agnostic Federated Learning
FL	Federated Learning
FTL	Federated Transfer Learning
GAN	Generative Adversarial Network
HFL	Horizontal Federated Learning
NbAFL	Noise before Agnostic Federated Learning
SGD	Stochastic Gradient Descent
SMC	Secure Multiparty Computation
VFL	Vertical Federated Learning

Chapter 1

Introduction

1.1 The Concerns Regarding Prostate Cancer

Despite having many clinical research and attempts of innovations, Prostate cancer is the second most common uncontrolled disease (after lung cancer) in men worldwide, counting 1,414,259 new cases and causing 375,000 deaths worldwide where 7.3% of all deaths caused by cancer in men in 2020. People from regions like Northern America, Northern and Western Europe, New Zealand and Australia face higher incidence rates than the people from regions like Northern Africa and Asia sustaining cases 37.5 and 11.3 per 100,000 respectively [1]. Besides, in 2021, the American Cancer Society's estimate for prostate cancer in the United States are about 248,530 new cases and about 34,130 of whom may die from prostate cancer [2]. Moreover, it is expected that upto 1.7 million new cases of prostate cancer, along with 499,000 new deaths caused by it will occur by 2030 simply due to the aging population of the world [3].

The majority of prostate cancers develop slowly, however some cancer glands grow quickly and spread from the prostate to other parts of the body, particularly the lymph nodes. It is difficult to forecast the early stages of this dangerous prostate cancer since it has few symptoms such as constant urination, hematuria, urine stream, dysuria, release of fluid in the prostate, bone pain in the femur, and so on. Prostate cancer has a significant risk factor because of its limited symptoms, such as obesity, genetic alterations, family history, age, and medical history [4]. There have also been cases where prostate cancer continues to grow despite there being incredibly low testosterone in the bloodstream due to the hormonal treatment or medical conditions. Different testing techniques [5] have been used to predict prostate cancer, but because these methods take a long time, it is difficult to come up with a decision that will be more advantageous in a short amount of time.

1.2 Machine Learning in Healthcare

Machine Learning has become an increasingly promising approach for patients' healthcare. [6] However, all of these schemes are entirely data driven. For cases where centralized ML schemes are used, where all the data are pooled into a central server, patients' privacy compromise has been a big issue in terms of data exchange of medical imaging records. Due to the lack of proper decentralized methods, a patient's privacy might be compromised which is very much held in contempt in case of medical information [7]. However, this problem can be solved by implementing a secure and private AI based decentralized and federated data storage and learning system replacing the present pattern of data sharing and centralized storage, which has a higher risk of compromising individual privacy. At present, patients' private data, despite needing to be protected, are mostly left exposed to all digital influences under current data sharing systems, meaning, such unregulated use of such private data may lead to cases of misuse — notably for financial gain, which may increase day by day, or other nefarious purposes. Our solution is Federated Learning, which we believe may be much more reliable in terms of keeping medical records safe and secure.

1.3 Federated Learning and Prostate Cancer

Prostate cancer, becoming an increasing concern for modern day men, requires extensive collaborative research and classification techniques. Naturally, the privacy of such personal medical conditions would need to be kept when performing the collaborative initiative for the intended robust and accurate classification system. In the proposed federated approach, all personal data regarding would be kept under the supervision of the institution they themselves are involved with and only locally trained models would be shared throughout the network for an aggregated, collaborative, robust and accurate model to be used globally. Such an approach, we believe, would be increasingly likely to be used not only in medical imaging but also in other data security platforms.

1.4 Research Motivation

According to [8], previously published federated learning systems only focused on optimizing a specific problem, such as security, privacy, accuracy, communication efficiency, or a broad unclassified sector. By employing LoAdaBoost, FedAvg, there were concerns (computational cost, computational complexity, and test accuracy) that were targeted to keep in mind; yet not for any particular sector in the same research. However, no previous published research has taken place in the medical field where security, privacy, test accuracy and communication efficiency were all taken into account. This study came up with an approach for detecting prostate cancer and classifying it using a decentralized, distributed, and safe federated learning system, taking into account data privacy, security, test accuracy, and communication efficiency for a specific medical sector. Furthermore, this study may be able to detect deadly prostate cancer in a timely manner, thus saving some lives.

Chapter 2

Research Problem and Objective

Federated learning entails training statistical models in expansive heterogeneous networks of remote devices with localized data which requires overcoming some unorthodox challenges —

2.1 Effect of privacy concerns

Most cases of medical imaging data are kept isolated with supplier institutions and for this reason, a large amount of dataset is needed to be transferred. As this type of transfer causes more and more legal challenges over a patient's privacy, it is very difficult to find enough medical image dataset.Because of few medical imaging datasets, models which were trained by a few of these datasets often fail to work on different protocols, equipment and patient population. But using FL, data can be used in different institutions to train the DL model by distributing training operations without using a combined single dataset [9].

2.2 System Diversity

Different federated learning devices may have different storage, size, analytical, and connection capabilities, and these capabilities may result in varying hardware, network, and power connectivity. Thus, while building or analyzing through federated learning, it is important to keep a few things in mind: accepting diverse hardware, anticipating a limited number of involvement, and being strong to network-released devices [10].

2.3 Expensive Communication

In federated networks, communication is a key constraint, which, along with privacy issues about transferring raw data, mandates that data generated on each device remain local. Besides, in federated learning, many devices are required to train or communicate, and network connectivity can be slower than local computing in various ways of values. It is critical to develop effective communication mechanisms that will frequently communicate model updates as part of the training process rather than sending the entire dataset over the network. Among local updating methods, primal-dual methods are found to be famous for convex objectives and distributed local-updating primal methods have practically shown immense improved performance over traditional approaches for non-convex objectives [10].

2.4 Synchronous or asynchronous training algorithm

The choice of whether to focus on asynchronous or synchronous training algorithms is a fundamental design decision for a Federated Learning infrastructure. But depending on the training algorithm, there might be some differences. Some successful work has been done using asynchronous training e.g. [11] and to address the issue of systems heterogeneity, asynchronous communication is popular to alleviate untidiness but can be irrational in federated learning due to the possibility of unlimited delay. However, according to [12] currently the synchronous training algorithm is highly in use even in the data center. Because some notion of synchronization is needed on a finite amount of devices so that a simple portion of the updates from multiple users is consumed less in the learning phase of the algorithm.

According to Li [10], to overcome these challenges various approaches can be taken. In addition, the Federated Averaging method is highly used for non-convex problems in federated settings. Among compression schemes, sparsification, subsampling and quantization can substantially reduce the space required for communicated messages per round. However, in federated settings, it is challenging to implement these compression schemes directly. On the other hand, star topology is the preeminent communication topology for decentralized training and it is found to be faster than centralized training in data center environments. Moreover, it can theoretically reduce communication expenses on global servers in federated learning but in practical as it has limitations training over heterogeneous data, hierarchical communication patterns are proposed to reduce load on global servers.

2.5 Research Objective

The goal of this research is to make the detection of prostate cancer easier, less time consuming and protected using Federated learning. Since FL is a distributed and decentralized machine learning method that allows for the training on the large substances of decentralized data residing in devices, it can be used to train the medical image dataset and patient's privacy can be maintained. The following are the objectives of this research:

1. To gain a deeper understanding of federated learning and how it works.

2. To gain a thorough understanding of how Federated Learning may be used to maintain privacy, as well as how it may be applied.

3. To create a model for detecting prostate cancer malignancy and test it on a variety of medical datasets.

4. To evaluate whether Federated Learning is a more reliable and secure approach in handling the classification of prostate cancer.

5. To offer possible improvements on the model for better utility and/or security.

Chapter 3

Literature Review

In 2020, prostate cancer being the fifth leading cause of cancer deaths among men [1], raised the necessity of its early detection for immediate treatment. In this case, deep learning is a potent approach to deal with this emergency as many works in the field of medical imaging diagnosis have been accomplished using deep learning. However, in order to reach clinical-grade accuracy, deep learning models require training from substantially huge curated datasets but medical data is supremely confidential and its usage is highly restricted [6]. As a result, collecting medical data is a very difficult and time consuming process. So, to deal with this issue, a secured deep learning approach can be adopted which is known as federated learning.

Kelloff and co. [13] mentions prostate cancer being the second most common cause of cancer-death among American men and not being invariably lethal. Current standard imaging techniques, such as ultrasound, MRI, CT and nuclear medicine cannot detect them in early stages. However, Song [14] mentions how the technology has improved over the years, coining the term mp-MRI (such as T2-weighted imaging, diffusion imaging, and dynamic contrast-enhanced imaging for lesion detection, differentiation and staging). There is no doubt that these sorts of imaging plays a pivotal role in various stages of cancer treatment. That being the case, it is crucial that ways to reliably and efficiently classify them become common practice.

Rieke, Hancox and their colleagues [6] mention how Federated Learning will become a huge contributing factor in managing and handling the health of the populace digitally. Their implementation methodology involves the Hub Spoke (a form of centralised topology) with the FedAvg aggregation approach, similar to our proposal. The peer-to-peer approach was mentioned on an occasional basis, whereby the challenges would be greater compared to the centralised approach. Challenges in their implementation involved data heterogeneity (multi-modality, multi-dimensionality and multi-characteristic data), privacy security (trusted or non-trusted collaboration where strict encryption systems may be needed to be placed), traceability accountability (the need for measuring contribution and quality from the plethora of different clients participating in the endeavour) and system architecture.

The authors Nilsson A, Smith S, Ulm G, Gustavsson E and others discuss an algorithm called FedAvg in their study [15], which works with a server-client relationship where the server initiates training. The server sends a global model to all of its clients, and the Stochastic Gradient Descent Algorithm is used to optimize it at client level. The FedAvg algorithm takes into account a total of five hyperparameters: learning rate, batch sizes, number of epochs, number of customers, and learning rate decay. Clients split their own data into different batch sizes and complete a certain number of SGD epochs after adjusting local models to shared models. The updated models are transmitted back to the server by the clients once the local models have been adjusted from the global model. Upon receiving the models from clients, the server brings changes to the global model by enumerating the weighted sum of all the acquired local models. Despite its effectiveness, the FedAvg algorithm still falls short in circumstances like the case of device heterogeneity. One of the prevalent problems is when a device fails to maintain the task of calculating a given number epochs within the time restriction and gets dropped by the server.

Choudhury and co. [16] address the many practical challenges when dealing with machine learning in real-world health data. Their main methods involve federated learning as the main framework upon which the work is implemented, along with differential privacy which is used to guarantee the privacy of the algorithms working on aggregated data. Three classification algorithms were used (perceptron, support vector machine and logistic regression) and in order to evaluate the models, both before and after performing the privacy-preserving mechanism, their utility is measured in terms of F1 score. 70% of the data was used to train the model with 5-fold cross-validation. 10 sites (clients) were used in the experiment on high grade server CPUs and RAM; done on 10 rounds of iteration. The last phase includes comparing and contrasting between centralized learning, FL, and FL with -differential privacy in terms of utility, for   between 0.01 and 0.5. Results showed that FL archives comparable performance to centralized learning for ADR and mortality prediction. Additionally, despite differential privacy guaranteeing a set level of privacy according to the set parameter, it led to a significant drop in performance and utility.

Another paper, also done by Choudhury and co. [17] focuses on, primarily, the privacy preserving aspect of federated learning in general, where they propose a syntactic approach for offering privacy in the context of FL. The goal was to increase data value and model performance while maintaining a verifiable and defensible level of privacy that met privacy legal framework requirements. The experimental results showed a consistent improvement for larger values   affecting utility positively across all classification algorithms. Additionally, the parameter k used for their syntactic approach yielded more generalized forms of their datasets for higher values to create equivalence classes, thus affecting the utility negatively; a behaviour common across all classification algorithms. Meaning, since their approach is heavily focused on privacy, their results were as expected, in that, it was a trade-off one needed to take.

Xu, J. and co.'s paper regarding federated learning for healthcare informatics [18] also addresses the issue of medical records not being properly utilized for robust and better healthcare due to the fragmentation and sensitivity of the data, and also how federated learning offers to be a completely unutilized solution to the problem. It also addresses the common challenges faced when taking a naive, or unsophisticated approach in the implementation of federated learning. The most common solution is to force the data to adapt to a uniform distribution, such as, using the minimax optimization scheme of agnostic federated learning (AFL) [19]. Alternatively, we could also share a common set of data globally across all clients [20][21]. The entire paper acts as a general summary of different federated learning algorithms, techniques and challenges, and provides a wide range general solution to them, as well as a survey of a set of representative FL methods for healthcare.

Although federated learning is focused on preserving the privacy of clients' data in a distributed machine learning setup, it is not impossible for information leakage to nor occur. Wei, K et al [22] mentions in their paper about Federated Learning with Differential Privacy that private information divulsion can occur by analyzing the uploaded parameters from the clients, and proposes a framework where artificial noise is added to the parameters at the clients' ends before aggregation of the model takes place (NbAFL). Proof of their framework working is attempted under certain circumstances, whereby their theoretical convergence bound reveals key properties, such as trade-offs between convergence performance versus privacy levels, and there being an optimal number of communication rounds for convergence performance for a fixed level of privacy. Their evaluations are proved to be consistent and obey their design principles.

HybridAlpha [23] is another such framework proposed by Xu, R and co. which focuses on dealing with the drawbacks of standard frameworks, such as differential privacy like the above and secure multiparty computation (SMC), that have large communication overhead and slow training time. HybridAlpha employs an SMC protocol based on a kind of functional encryption, which allows it to be simple, efficient and resilient to unforeseen circumstances like clients canceling in the middle. Evaluation procedures show that HydribAlpha reduces training time by 68% and data transfer volume by 92% on average, all the while maintaining the same level of performance and privacy as existing solutions.

The usefulness and applications of federated learning can also be found in other fields as well, such in mobile networks [24] and the industrial areas [25]. In mobile networking, virtual keyboards such as Gboard and SwiftKey use FL to improve language suggestions, service providers can use models that employ FL that use location history of mobile devices to accurately provide entertainment and restaurant recommendations, delivery and commute services can use real time traffic information to provide accurate ETAs for arrival and travel times, and healthcare data can be shared among hospitals or medical researchers to improve clinical services and analytics; all the while making sure the actual data that is used to train the different types of models stay in the phone they're trained from.

Truex, S. and co. [26] touch upon issues such as maintaining data locality on its own does not provide sufficient privacy guarantees and that an FL system that can prevent interference both during training and final trained model needs to be made. Similar to Xu, R's paper on HybridAlpha [23], they also establish the drawbacks in accuracy of systems such as SMC, as well as trying to develop their own differential privacy system that is both well-balanced in terms of security and accuracy, as well as being scalable vertically (more complex models) and horizontally (various kinds of models). In addition to employing differential privacy, they also add in a threshold homomorphic encryption algorithm such as the Paillier cryptosystem [27].

Yang, Q., Liu, Y., Chen, T., and Tong, Y. [28] propose several types of federated learning frameworks such as horizontal FL, vertical FL, and federated transfer learning, as well as provide architectures, applications and surveys on the topic. HFL is used in cases where datasets share the same feature space but different space in samples, for example firms providing the same kinds of products/services can have the same feature space, but may have different customer groups with very little intersections, so different sample space. The problem with this type of model is that it assumes honest participants and the server is honest-but-curious i.e. only the server can compromise on the data privacy, which may not work in certain real life scenarios. VFL is the opposite, where two datasets can share the same sample space but different feature space. Here, attackers can only learn about the data from the infected client, and not from other clients beyond what is revealed by the parameters that are relayed in and out. In FTL however, both spaces differ. This might seem like two irrelevant datasets are being used in a single federation, however, the point here is to actually find and learn about any common representation between their feature spaces.

From the above discussion, we can infer that federated learning has the potential to be a feasible approach for early medical imaging diagnosis ensuring patient confidentiality but there are aberrant challenges that need to be subdued. Albeit there have been attempts made such as differential privacy technique, syntactic anonymization to address the issues of data privatization it has also led to degradation in prediction accuracy in many of the cases. Along with this, the heterogeneity in devices and their computational ability comes in the way of accomplishing the desired number of epochs and model encryption in time. Furthermore, another common problem that is encountered is the variance of data availability at client level that may not have adequate data for the deep learning models to be well suited. All of these shortcomings leave the door open to work in these aspects to look for models that will yield better accuracy and be well suited for data related to prostate cancer.

Chapter 4

Work Plan

4.1 Training Phase

Our work plan involves a standard machine learning model that takes in medical imaging as training and testing data. Initially, the model is trained on the main server using open source imaging data as a preliminary form of fitting.

Next, in order to simulate multiple computers from different hospitals, the updated model is sent to each of these clients, where separate and independent imaging data is stored. Here, each client uses their own data to further train the model locally, resulting in n different versions of the original global model, where n represents the number of different clients participating in the training phase.

Once the training is done, the local models, and only the models, are sent back to the main server. The data that was used to locally train the copies of the model remains with the clients. The collected models are then collected and aggregated into a single, more sophisticated global model, which is then redistributed to the clients for further use.

This training may be repeated several times in order to make sure the model is not underfitting.

It must be noted that the image training is done on only one 'type' of imaging data, such as MRI or CT scans; not both at the same time. A different model needs to be used for new types of scans.

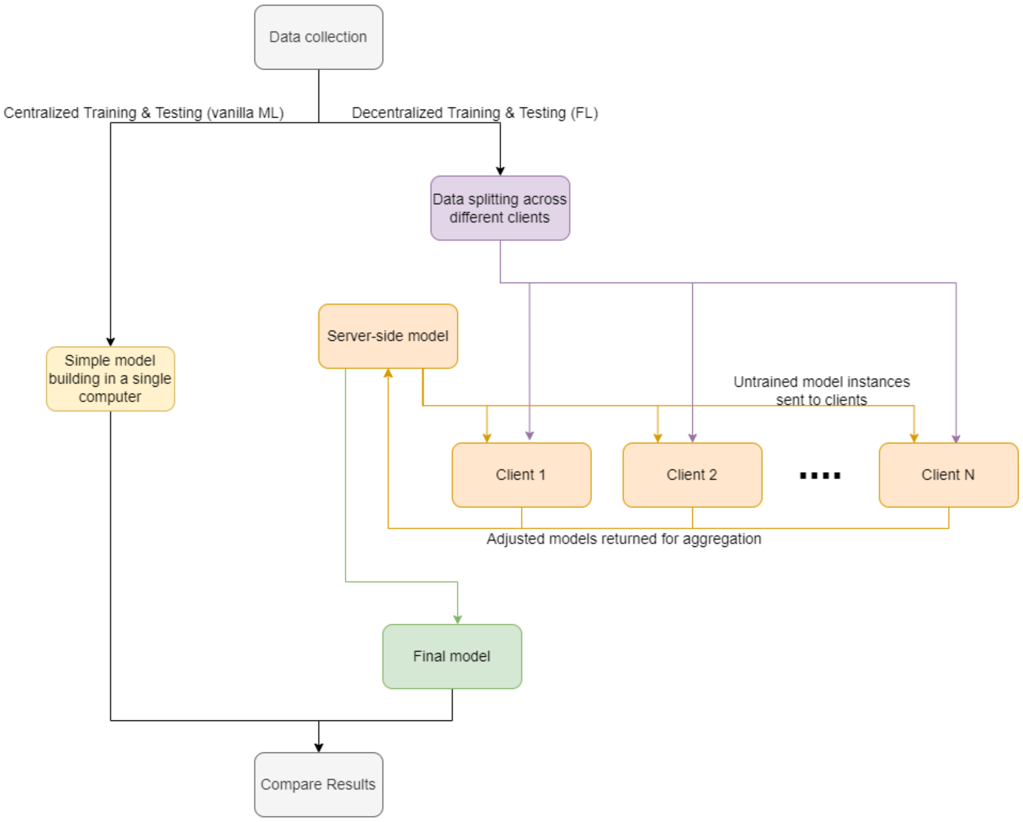

Data collection
Centralized Training & Testing (vanilla ML)
Decentralized Training & Testing (FL)
Data splitting across different clients
Server-side model
Simple model building in a single computer
Untrained model instances sent to clients
Client 1
Client 2
....
Client N
Adjusted models returned for aggregation
Final model
Compare Results

Figure 4.1: Workplan

4.2 Testing Phase

Before deployment, the trained model is tested on the server side with a different set of images. The model outputs the results with a certain degree of confidence (e.g. 95% confidence in malignancy, 80% confidence in non-malignancy, 75% confidence in unsurety). The confidence values are simply the output numbers of the model converted to a percentage. This may also be represented as a ratio. In cases where the confidence differences between the three outputs are too close (such as being 95% malignant and 80% non-malignant) the model is passed through the training phase once again, possibly with a different set of data.

If the confidence differences are large enough (98% malignant, 37% non-malignant, 2% unsure), it would mean the model passed the testing phase and is ready for deployment.

4.3 Post-Deployment Phase

After the model is deployed into the real world, it still does not stay static. Since machine learning is all about constantly updating and improving the model, it must go through an indefinitely continuing training and testing routine.

Here, data from a new patient is put into the model (local), supposedly by the doctor or technician etc. involved in the patient's diagnosis. The model outputs the results of the imaging, stating how confident it is in the malignancy of said tumor. If the doctor approves of the results, that is, deems the confidence sufficient enough to be a reliable outcome, then this output is used to fine-tune the weights of the model for stronger confidence.

If the doctor disapproves of the result, meaning, it is deemed not sufficient enough to be a safe and reliable outcome, the doctor manually looks into the imaging to determine the malignancy of said tumor. After this, the doctor's result is inputted into the model which is used to retrain it for better results. Ultimately, the model must go through continuously updating phases.

After these steps are carried out in all the clients involved, the locally updated models are sent back to the server at the *scheduled time*. No further updated models will be accepted into the server after this time is up. The aggregated global model is then redistributed across all clients to be used by the doctors once again, hopefully, for better future diagnoses and classifications.

Chapter 5

Methodology

The suggested prostate cancer classification model's goal is to detect prostate cancer in data separated by PI-RADSv2 grades. To accomplish so, the model needs to create a mechanism that receives data from files of prostate cancer patients as input, processes it systematically, and produces predictions of four types: PIRADS1, PIRADS2, PIRADS3, and PIRADS4.

5.1 Dataset

Most medical data is immensely confidential and used under extreme restrictions [4]. So the fundamental initiative of our research was to collect a credible Multi-parametric MRI dataset that we were able to download from "The Cancer Imaging Archive" [29]. The dataset was created by conducting the study on old men aged 47-69 years, based on data gathered in a single-center condition from PCa 3 T mpMRI acquisitions in a treatment-naive population employing an endorectal coil under standard-of-care clinical conditions [30]. In fact, it comprises data of double mpMRI examinations of 15 patients each. With no intervening treatment, imaging was done at intervals of up to 2 weeks [30]. In this dataset, three types of scanned images are available. They are:

- Axial T2-Weighted (T2AX)
- Diffusion-Weighted or Apparent Diffusion Coefficient (DW or ADC)
- Dynamic Contrast-enhanced Subtract (DCE or SUB) [30].

T2AX imaging technique performs best in detecting prostate cancer in the transitional zone, DW or ADC imaging technique performs dominantly in detecting prostate cancer in the peripheral zone and only a qualitative assessment with the presence or absence of focal enhancement is recommended by using DCE or SUB imaging technique [31]. Among these three types of scanned images, we have chosen to work with the Axial T2-Weighted images. The reason behind selecting T2AX images is, 8 out of 11 patients who were diagnosed with tumor, had a Tumor-suspicious Region Of Interest (tROI) volume of less than 0.5 mL which was derived from averaging between the calculations acquired in baseline and repeat T2AX images whereas usually clinically significant prostate cancer patient has tROI volume of greater than 0.5 mL [30]. Therefore, we also took into account the T2AX image data of baseline study as well as of repeat study for training the model. The file format of the images are .dcm (DICOM images) and there are on average 30 T2AX dicom image slices in each study of each patient, so on average there are 900 T2AX dicom slices.

For labelling the dataset, we have selected the PI-RADS scoring system that strives to ensure that mpMRI findings are consistently interpreted, communicated, and reported [30]. PI-RADS score ranges from 1 to 5 but in our dataset we labelled classes from only 1 to 4 because there was no patient recorded with PI-RADS score 5 in baseline-repeat study. The PI-RADS score denotes the likelihood of the presence of clinically significant prostate cancer, the interpretation is given in the following:

- PI-RADS 1 – Quite Low (High possibility of clinically insignificant cancer being present)
- PI-RADS 2 – Low (Possibility of clinically significant cancer being present)
- PI-RADS 3 – Medium (Uncertain presence of clinically significant cancer)
- PI-RADS 4 – High (High possibility of clinically significant cancer being present)[31]

5.2 Data preprocessing

5.2.1 Centralized

The next step was processing the images of patients using different mechanisms. It is very important for the images to be in the same size to be able to feed in a convolutional neural network. All the MRI scans were converted from dicom files to image data. The data were then divided into 4 folders depending on their PI-RADS value. Consequently, the data were randomly splitted into train (70% of the images), validation (20% of the images) and test (10% of the images) set. This split resulted in 624 images belonging to train samples, 179 images belonging to validation samples and 89 images belonging to test samples. The batch size of the input data was set to 10 per batch. The images were then processed with the preprocess_input function of the VGG19 model and Xception model. For the VGG19 model, the image target size was set to 224 by 224 and the images were converted from RGB to BGR format in the function. Subsequently, with regard to ImageNet, the function would make the images zero-centric without scaling for each color channel. For the Xception model, the image target size was set to 299 by 299 and the pixels of the images are scaled in the range of -1 to +1 by the preprocess_input function of the Xception model. The classes defined for both the VGG19 and Xception models were PIRADS1, PIRADS2, PIRADS3 and PIRADS4.

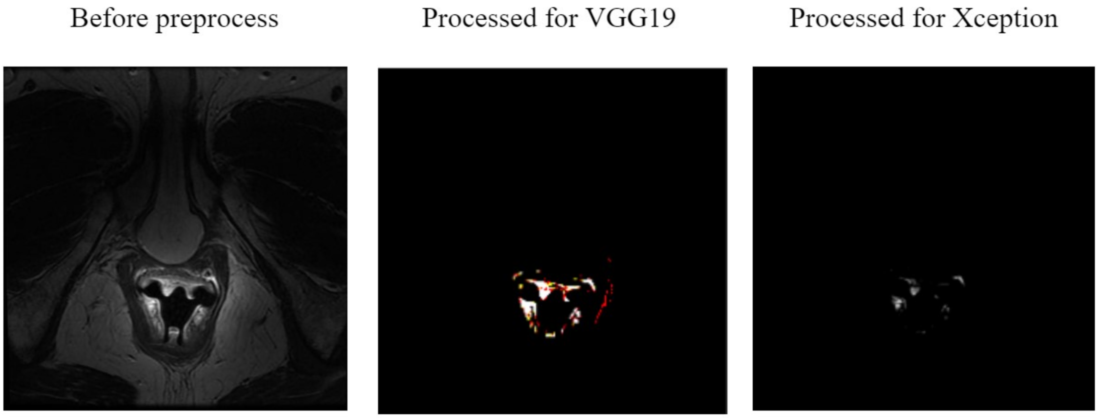


Figure 5.1: Visual comparison among pre-processed, processed for VGG19 and processed for Xception

5.2.2 Decentralized

The preprocessing for decentralized training started with taking in all the images together from the directories train,test and validation. The labels for the images were saved as numerical values such as zero representing PI-RADS score 1. The images were then converted into numpy arrays and resized to 224 by 224. Subsequently, the data were shuffled for well distribution and normalized to make sure the comparisons between data collecting techniques and texture instances are as accurate as possible. The data were splitted for train and test purposes with a ratio of 80 to 20 respectively. At this phase, the train data was distributed among the clients in a dictionary keeping the client names as keys and the data shards as value pairs. Following the data distribution, each client's data were batched to a size of 32 with the help of from_tensor_slices(). The same was done for the test data as well.

5.3 Experimental set up

The classification process of the MRI data is no different than any other image classification procedure after the format is changed from dicom files to jpg files. The image data are taken as input and processed to suit the specific algorithm that is being used. The labels are defined for each image data if we are following a supervised approach. However, if the unsupervised procedure is followed then it may not be necessary. The task of extracting attributes is handled by the neural networks.

5.3.1 Centralized architecture

In centralized architecture, we trained the models with data located in a central device. The training phase started by preprocessing the input image data. Next, the salient features were selected by the model via reducing the extracted features from the dataset. Then, the dataset was splitted into a train set and test set for learning and prediction purposes of the model respectively via neural networks leading to a classified result.

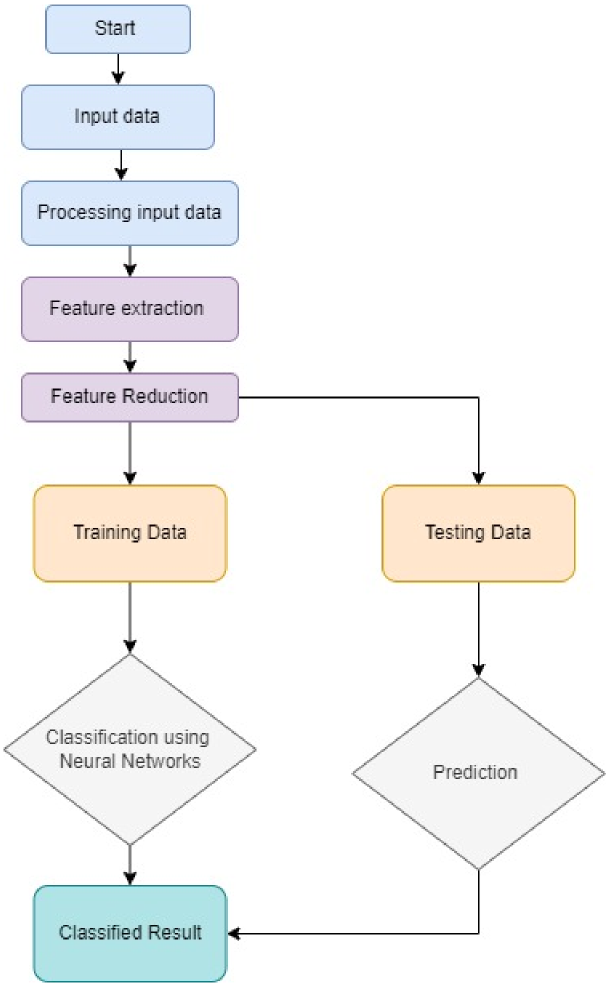


Figure 5.2: Dataflow of classification

5.3.2 Decentralized architecture

As the images were processed to the desired level, we created the deep neural model to initiate the training phase. We constructed two instances of the model, one to be globally used and one to be locally used by the end devices. An environment was created to be run for the set number of communication rounds which implies how many times the global weights are sent back to local to train. At each round, a random number of clients(end device) was picked to be trained. Afterwards, the clients were shuffled and the picked number of clients were selected and proceeded to be trained. While iterating through the clients, each client's model weights were to the global weights and trained for one round. Upon the finishing of a client's model train, the proportion of the client's training data was compared to the overall training data maintained by all clients to determine the scaling factor. The client's trained model weights were scaled using the value of the scaling factor generated in the previous step. These scaled locally trained model weights were saved to aggregate to the global model weights. Subsequently, at global level, the averaging was done on the scaled values component wise based on the percentage of the data points supplied by each participant client. Finally, the averaged values are set as the new weights for the global model and kept to be used at local level. The process repeats itself for the set amount of rounds.

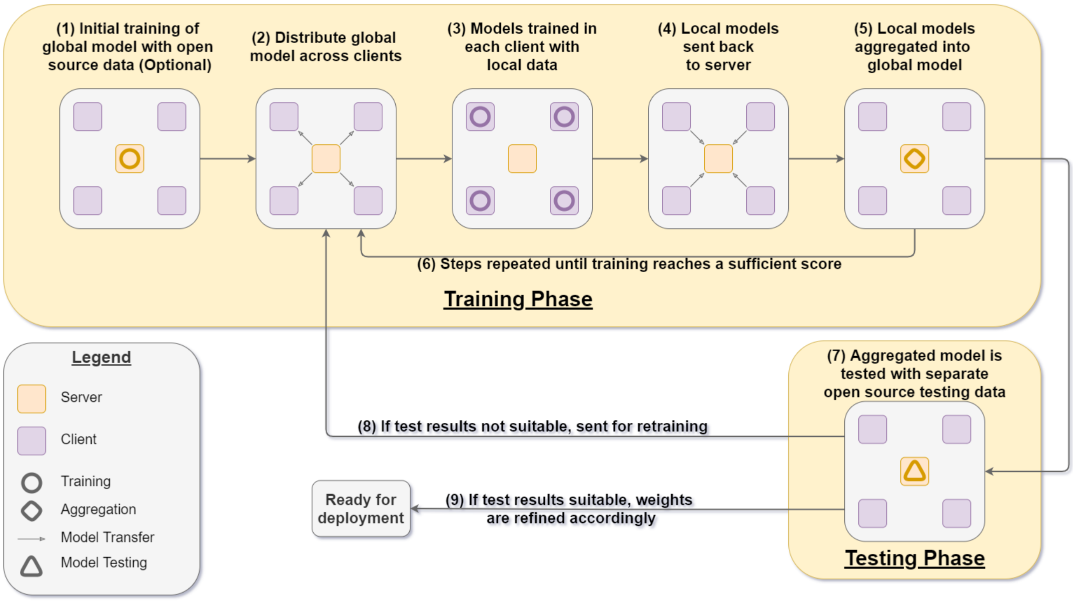

(1) Initial training of global model with open source data (Optional)
(2) Distribute global model across clients
(3) Models trained in each client with local data
(4) Local models sent back to server
(5) Local models aggregated into global model
(6) Steps repeated until training reaches a sufficient score
Training Phase
Legend
Server
Client
Training
Aggregation
Model Transfer
Model Testing
(7) Aggregated model is tested with separate open source testing data
(8) If test results not suitable, sent for retraining
Ready for deployment
(9) If test results suitable, weights are refined accordingly
Testing Phase

Figure 5.3: Decentralized model architecture

We have used one simple Convolutional Neural Network model and two state of the art pre-trained image classification models for classifying our MRI data. These three models were trained both in the centralized and the decentralized system for 50 epochs separately.

5.3.3 Simple CNN

The simple CNN model that we used was built on a Sequential class stacked with 3 layers of Conv2D network which had a gradually increasing number of filters 32, 64 and 128. These Conv2D layers help extract the features from input images. The activation function we used in all the Conv2D networks was the ReLU function and the padding was kept the same for all the layers. In between the Conv2D layers there were three MaxPool2D layers which had a pool size of 2,2 and strides size of 2 for all the layers. The MaxPool2D layers were used to reduce the spatial dimension by taking the maximum value over an input window of pool size keeping the depth constant. Following the Conv2D and MaxPool2D layer, we used a Flatten function to turn the varying dimensional tensors to one dimensional. Subsequently, we utilized a Dense function with a softmax activation function and unit size of 4. The model was compiled with a learning rate of 0.0001 and the loss function was set to categorical_crossentropy. Furthermore, we used the Adam optimizer as it is computationally efficient in optimizing the cost function by finding out the best set of parameters.

5.3.4 VGG19

There are many different CNN architectures and among those, the VGG19 has been chosen for our work because of its simplicity. There is another CNN architecture which is VGG16. Convolutional layers, max pooling layers, and fully linked layers make up the VGG16 model. There are 16 layers in all, with 5 blocks and a maximum pooling layer in each block. Nevertheless, VGG19 is significantly better than VGG16 although it uses more RAM. Moreover, VGG19 gives more accurate predictions than VGG16. This network has been pre-trained using the ImageNet database which contains a large number of images. VGG19 is a 19-layer variant of the VGG model which contains 16 convolution layers, 5 MaxPool layers and 3 Dense layers. In VGG19, as the filters are being increased, the convolution layers must be increased. More specifically, if in layer-1 there are 16 filters, other layers must have 16 or more filters. In our work, we excluded the prediction dense layer that was included in the model and added our own dense layer with the unit size of 4 and a softmax activation function. We used the Adam optimization algorithm with a learning rate of 0.00001 and the loss parameter was set to categorical_crossentropy. However, for the decentralized approach the learning was set to 0.0001.

5.3.5 Xception

Xception is an intermediate step between depthwise separable convolution and standard convolution (after a depthwise convolution, a pointwise convolution). A depthwise separable convolution, also known as Xception, can be thought of as an Inception module with a maximally significant number of towers. Inception modules were replaced with depthwise separable convolutions in Xception. In a nutshell, Xception's architecture is made up of Depthwise Separable Convolution blocks and Maxpooling, which are linked together by shortcuts of ResNet-style. A convolution layer in Inception attempts to learn filters in a three-dimensional space that has two spatial dimensions (width and height) and a channel dimension. A single convolution kernel is charged with mapping cross-channel and spatial correlations at the same time. Inception modules often look at cross channel correlations using a series of 1x1 convolutions, which map the input data into three or four smaller 3D spaces than the original input space, and then map all correlations in these smaller 3D spaces using standard 3x3 or 5x5 convolutions. Moreover, according to [32], Xception is a convolutional neural network architecture based entirely on depthwise separable convolution layers and a hypothesis which states that the mapping of cross-channel and spatial correlations in convolutional neural network feature maps may be completely separated. Basically, the Xception architecture is a depthwise separable convolution layer stack with residual connections that is stacked in a linear fashion. There are 36 convolution layers and 4 MaxPooling layers in this Xception model. This Xception is more or less similar to Inception but the basic difference is that On the ImageNet dataset (for which Inception V3 was created), Xception marginally outperforms Inception V3, and considerably outperforms Inception V3 on a bigger image categorization dataset. The model's prediction dense layer has been replaced with our own dense layer which has a unit size of 4 and a softmax activation function. We utilized GlobalAveragePooling2D to apply average pooling to the spatial dimensions until each is one, while leaving the other dimensions alone with the parameter (None, 2048). Here, Adam optimization method has been used with a learning rate of 0.0001 and categorical_crossentropy as the loss parameter.

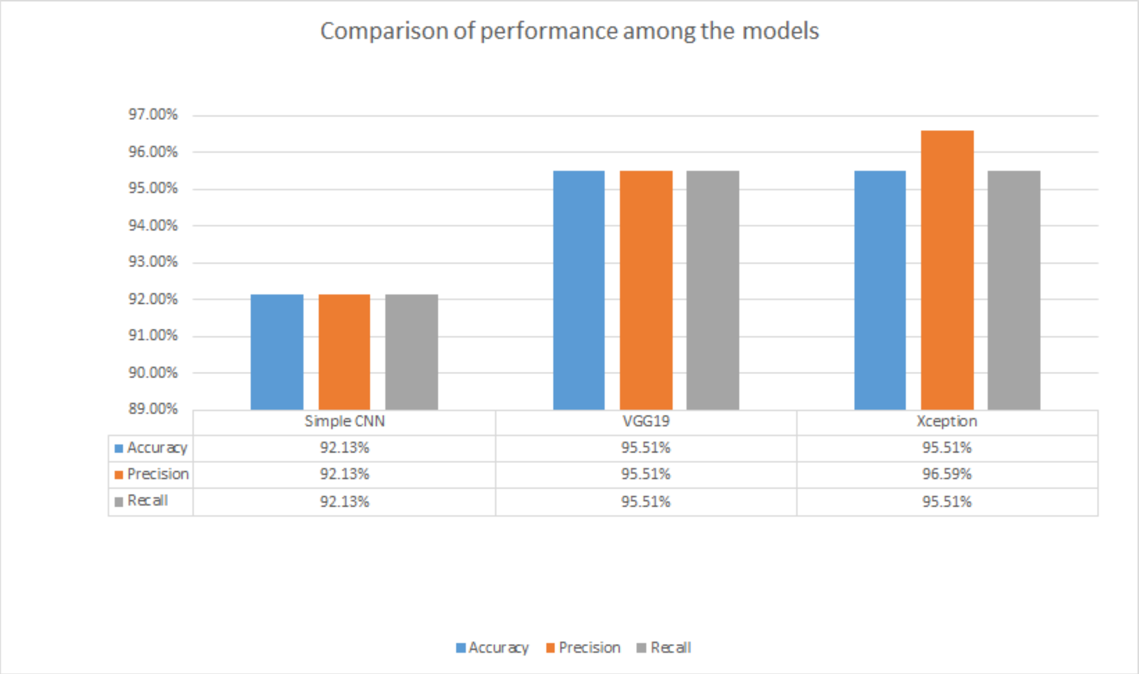
Chapter 6

Results & Analysis

6.1 Centralized

In this part, the performance of the three models that we used to classify the MRI data would be discussed. The simple convolutional model achieved an accuracy of 92.13%. The precision and recall value was also the same for this model. The VGG19 model yielded an accuracy of 95.51% although it had a validation accuracy of 97.21% while training. It failed to correctly classify 3 MRI data belonging to the PIRADS2 class and one belonging to the PIRADS3 class. Similar to our simple convolutional model, the VGG19 model also had the same precision and recall score as its accuracy rate. The similar ratio of true positive and alternating value of false positive and false negative among the four classes led to the equal value of precision and recall score. The Xception model produced an accuracy of 89.47% when the learning rate was set to 0.00001. However, the model was able to achieve the same accuracy of 95.51% as the VGG19 model when we increased the learning rate to 0.0001. The precision rate and recall value after the change in learning rate was 96.59% and 95.51% respectively.

According to the accuracy, precision and recall, the comparison among Simple CNN, VGG19 and Xception is shown below:

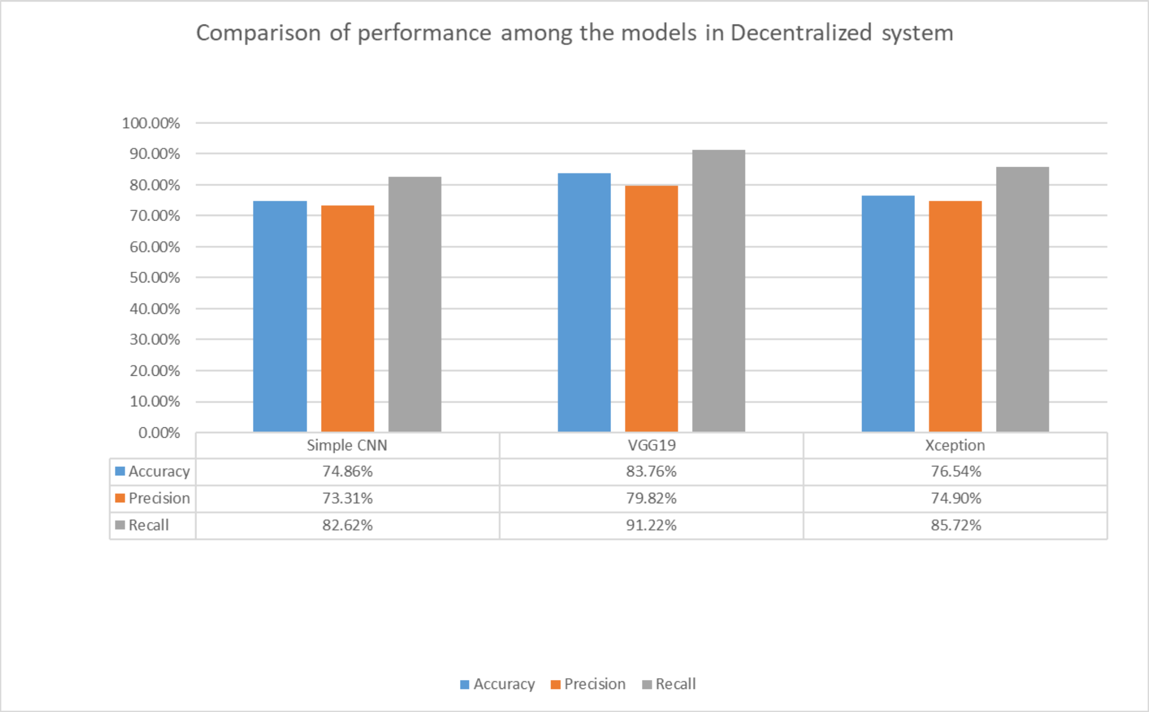


	Simple CNN	VGG19	Xception
Accuracy	92.13%	95.51%	95.51%
Precision	92.13%	95.51%	96.59%
Recall	92.13%	95.51%	95.51%

Figure 6.1: Comparison of performance among the models in centralized system

6.2 Decentralized

This section includes the performance of the same three models but in a decentralized environment i.e. federated learning approach. Carrying the same parameters like learning rate, epochs albeit the different setup, our simple neural network model could afford to reach an accuracy of 74.86% on the test data while having a recall and precision score of 82.62% and 73.31% respectively. The Xception model reached an accuracy of 76.54%. In the meantime, it gained a recall value of 85.72% and a precision value of 74.9%. Among the three of the models we tested, the VGG19 model stood out and reached an accuracy of 83.76% while having a recall score of 91.22% and precision score of 79.82%.



	Simple CNN	VGG19	Xception
■ Accuracy	74.86%	83.76%	76.54%
■ Precision	73.31%	79.82%	74.90%
■ Recall	82.62%	91.22%	85.72%

Figure 6.2: Comparison of performance among the models in decentralized system

6.3 Comparison between centralized & decentralized setup

6.3.1 Simple CNN model

While training the dataset on the CNN model in the centralized system our model saw quite a spike in learning for the first ten to twelve epochs and quite a decrease in loss function. For the rest of the epochs the model saw very little improvement. On the contrary, in the decentralized system, the model saw barely any improvement in learning for the first fifteen epochs. While the rest of the training period saw varying learning spikes. The model observed a decrease in the loss function consistently through the training period.

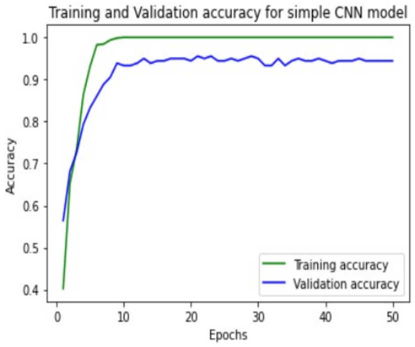
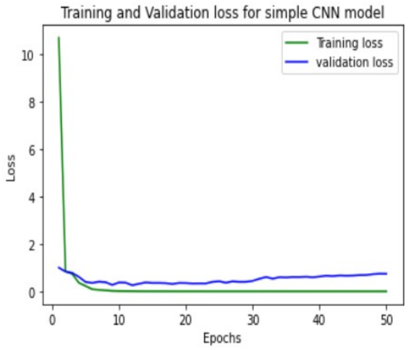
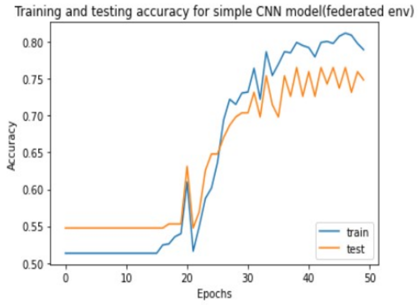
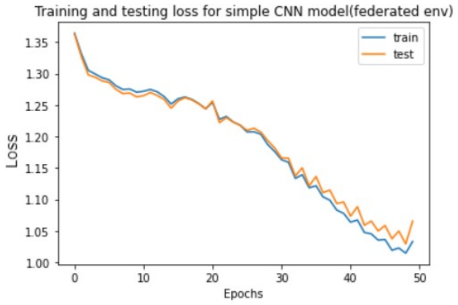
	Accuracy Histogram	Loss Histogram
Centralized		
Decentralized		

Figure 6.3: Accuracy and Loss Histogram of Simple CNN Model

6.3.2 VGG19 model

Our VGG19 model reached peak accuracy very quickly in the centralized system, before 5 epochs, whereas our decentralized model merely started to become more accurate at the same number of epochs, taking around 30 epochs to reach peak accuracy.

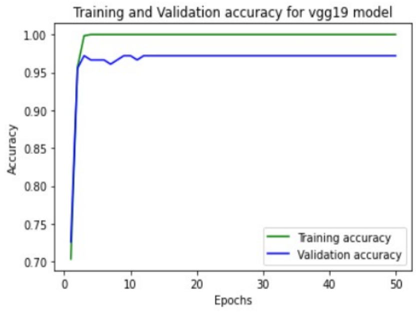
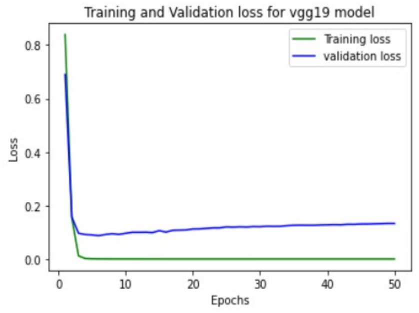
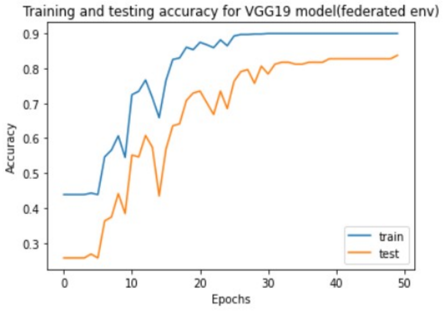
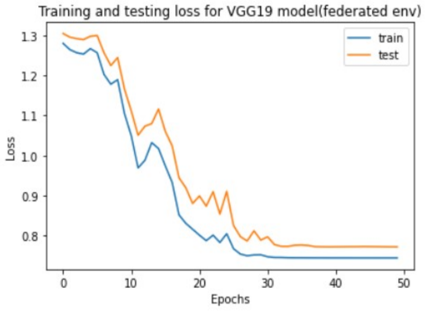
	Accuracy Histogram	Loss Histogram
Centralized		
Decentralized		

Figure 6.4: Accuracy and Loss Histogram of VGG19 Model

6.3.3 Xception model

Despite reaching peak training accuracy before 5 epochs, our centralized Xception model had very frequent and sharp highs and lows during its validation phase. Even so, the general trend was still heading towards a better direction. On the other hand, our decentralized model underwent a steady and gradual improvement throughout the 50 epochs it went through.

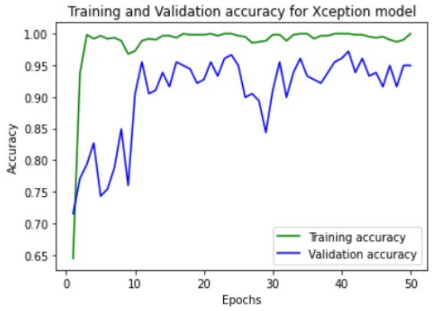
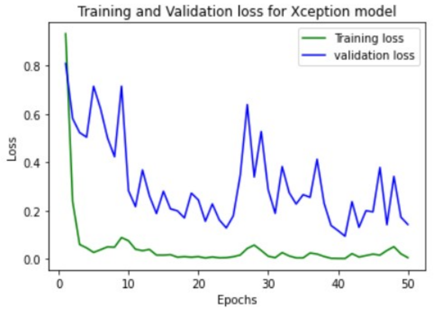
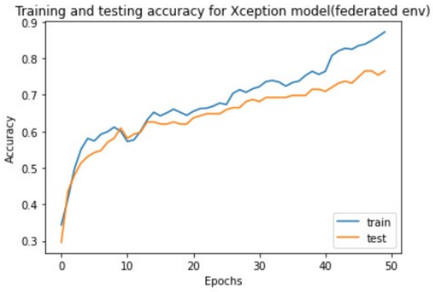
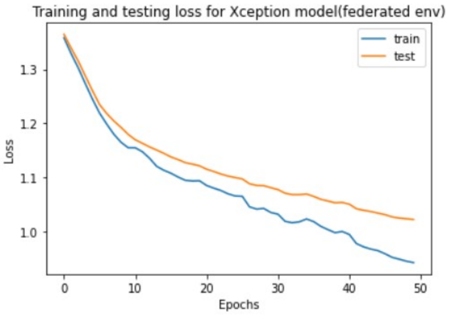
	Accuracy Histogram	Loss Histogram
Centralized	 Training and Validation accuracy for Xception model 	 Training and Validation loss for Xception model
Decentralized	 Training and testing accuracy for Xception model(federated env) 	 Training and testing loss for Xception model(federated env)

Figure 6.5: Accuracy and Loss Histogram of Xception Model

6.3.4 Overall Accuracy Comparison

Although the scores gained in the decentralized system could not exactly reach the scores gained in the centralized system running the equal number of epochs, our models produced promising results that can be further improved with the change in likes of communication round and feeding more data.

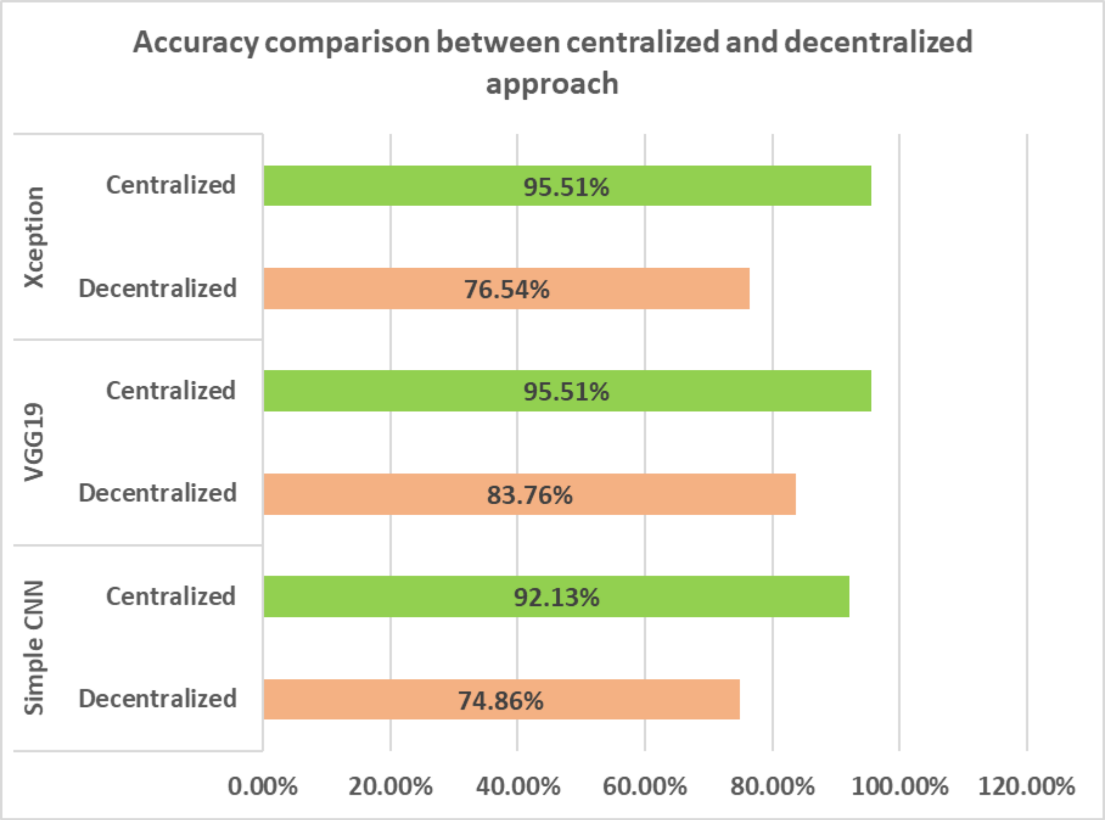


Figure 6.6: Comparison of performance among the models in centralized & decentralized system

Chapter 7

Conclusion & Future Work

7.1 Future Work

The goal of this study was to apply federated learning, which is a decentralized privacy-preserving learning method, to protect the privacy of personal medical data. Because medical data has such high privacy concerns and it is essential that the privacy of such data is to be protected, a technique that will provide considerable benefits in terms of privacy preservation is required. As a result, the decentralized learning approach of Federated Learning was used in this study. Even though it is a decentralized learning method, it still faces several obstacles, such as server-side threats or poisoning attacks [33]. In some situations, a poisoning attack in the decentralized system may allow user-level privacy to be compromised. In the future, a framework containing Generative Adversarial Networks (GAN) might be utilized to handle this problem. A multitask discriminator might also be added to the framework to allow the system to save private data requested by the user. This will differ from the current research procedures in some ways. The training process is limited under the current procedures, however this has no bearing on the conventional training mechanism. Using the mGAN-AI practical reconstruction attack, a malicious server may assure the reconstruction of genuine training data [34]. However, there are still some improvements that may be made in the future. Besides, in real life, decentralized data distribution may often be non-IID i.e. non-independent and identically distributed since different hospitals or clients may have local data of only one class or few classes out of many classes which can worsen the model's performance. For instance, if one client contains the data of patients with PI-RADS score-2 only, then the model may not learn to classify the images having PI-RADS score-1, 3, 4 .etc. Therefore, this challenge needs to be addressed and worked on. On the other hand, since it is a decentralized learning approach, in real life scenarios, data may get increased during the runtime of the training process as there is no centralized control over the clients, from the server and data of new patients may get added at any moment. Hence, further research is needed to address this challenge and to solve it in an appropriate manner.

7.2 Conclusion

Deep learning, particularly federated learning has resulted in a plethora of digital healthcare technologies [6]. To summarize, by utilizing federated learning, this research can be used to develop a safe and reliable approach for identifying prostate cancer in a short amount of time. In this research, the classification process was completed first, and then federated learning was employed to provide a secure and trustworthy approach for protecting medical data. Since privacy is a major problem, decentralized federated learning was used in this work, and the efficiency of the federated learning was tested using a set of medical image data, particularly for our purposes, from prostate cancer patients. The comparison between the efficiency of centralized and decentralized approaches have been shown in this research. However, if the necessity of this research is being discussed, it can be said that as a result of early detection, prostate cancer patients who strive to keep their disease hidden due to social stigma can begin their treatment sooner by assuring them with their privacy concerns while also preventing the illness from becoming aggressive.

Bibliography

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, “Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[2] R. B. Kargbo, *Androgen receptor degradation for therapeutic intervention of prostate cancer drug resistance*, 2021.

[3] M. M. Center, A. Jemal, J. Lortet-Tieulent, E. Ward, J. Ferlay, O. Brawley, and F. Bray, “International variation in prostate cancer incidence and mortality rates,” *European urology*, vol. 61, no. 6, pp. 1079–1092, 2012.

[4] P. M. Shakeel and G. Manogaran, “Prostate cancer classification from prostate biomedical data using ant rough set algorithm with radial trained extreme learning neural network,” *Health and Technology*, vol. 10, no. 1, pp. 157–165, 2020.

[5] J. L. Rowles, K. M. Ranard, C. C. Applegate, S. Jeon, R. An, and J. W. Erdman, “Processed and raw tomato consumption and risk of prostate cancer: A systematic review and dose–response meta-analysis,” *Prostate cancer and prostatic diseases*, vol. 21, no. 3, pp. 319–336, 2018.

[6] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, *et al.*, “The future of digital health with federated learning,” *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[7] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, “Secure, privacy-preserving and federated machine learning in medical imaging,” *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[8] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, “Loadaboost: Loss-based adaboost federated machine learning with reduced computational complexity on iid and non-iid intensive care data,” *Plos one*, vol. 15, no. 4, e0230706, 2020.

[9] K. V. Sarma, S. Harmon, T. Sanford, H. R. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. G. Raman, R. Kulkarni, *et al.*, “Federated learning improves site performance in multicenter deep learning without data sharing,” *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1259–1264, 2021.

[10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[11] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, *et al.*, “Large scale distributed deep networks,” 2012.

[12] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, *et al.*, “Towards federated learning at scale: System design,” *arXiv preprint arXiv:1902.01046*, 2019.

[13] G. J. Kelloff, P. Choyke, and D. S. Coffey, “Challenges in clinical prostate cancer: Role of imaging,” *AJR. American journal of roentgenology*, vol. 192, no. 6, p. 1455, 2009.

[14] Y. Song, Y.-D. Zhang, X. Yan, H. Liu, M. Zhou, B. Hu, and G. Yang, “Computer-aided diagnosis of prostate cancer using a deep convolutional neural network from multiparametric mri,” *Journal of Magnetic Resonance Imaging*, vol. 48, no. 6, pp. 1570–1577, 2018.

[15] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, “A performance evaluation of federated learning algorithms,” in *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, 2018, pp. 1–8.

[16] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, “Differential privacy-enabled federated learning for sensitive health data,” *arXiv preprint arXiv:1910.02578*, 2019.

[17] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, I. Sylla, Y. Park, G. Hsu, and A. Das, “Anonymizing data for privacy-preserving federated learning,” *arXiv preprint arXiv:2002.09096*, 2020.

[18] J. Xu and F. Wang, “Federated learning for healthcare informatics. arxiv 2019,” *arXiv preprint arXiv:1911.06270*,

[19] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 4615–4625.

[20] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019, pp. 1–7.

[21] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” *arXiv preprint arXiv:1806.00582*, 2018.

[22] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[23] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, “Hybridalpha: An efficient approach for privacy-preserving federated learning,” in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 13–23.

[24] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, “Reliable federated learning for mobile networks,” *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[25] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, “Efficient and privacy-enhanced federated learning for industrial artificial intelligence,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[26] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, “A hybrid approach to privacy-preserving federated learning,” in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 1–11.

[27] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *International conference on the theory and applications of cryptographic techniques*, Springer, 1999, pp. 223–238.

[28] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[29] A. Fedorov, M. Schwier, D. Clunie, C. Herz, S. Pieper, R. Kikinis, C. Tempany, and F. Fennessy, “An annotated test-retest collection of prostate multiparametric mri,” *Scientific data*, vol. 5, no. 1, pp. 1–13, 2018.

[30] A. Fedorov, M. G. Vangel, C. M. Tempany, and F. M. Fennessy, “Multiparametric magnetic resonance imaging of the prostate: Repeatability of volume and apparent diffusion coefficient quantification,” *Investigative radiology*, vol. 52, no. 9, p. 538, 2017.

[31] P. Steiger and H. C. Thoeny, “Prostate mri based on pi-rads version 2: How we review and report,” *Cancer Imaging*, vol. 16, no. 1, pp. 1–9, 2016.

[32] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[33] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, “Poisoning attack in federated learning using generative adversarial nets,” in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, 2019, pp. 374–380.

[34] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, “Beyond inferring class representatives: User-level privacy leakage from federated learning,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, IEEE, 2019, pp. 2512–2520.