“Federated Learning for healthcare: analysing the recovery of covid-19 patients with long-term effect”

By

Usman Shoukat

Student ID: 201537600

This dissertation is submitted in partial fulfillment of the requirements for the award of degree of

MASTER OF SCIENCE IN DATA SCIENCE AND AI

Submitted to

University of Liverpool, England, United Kingdom

Primary Supervisor: Pamela Lins Bezerra

Secondary Supervisor: Rida Laraki

Abstract

Covid-19 has threatened and affected the health, life and productivity of humans. It is taking an adverse turn with people not recovering from covid over several weeks and months. AI can plays an important role in classifying and predicting severe cases so that they can be identified and cured earlier. Machine learning and AI algorithms require data on which models can be trained. The data relating to the health sector is very private and is not usually shared, thus leading to a scarcity of data to train the ML and AI algorithms. This data privacy problem can be solved by using federated learning in which the model is trained on the local data in the local centres, and in the end, all the local models are aggregated by some aggregation mechanism to the central model. In this way, data never leaves the local centres, and the model is trained on the available data without privacy concerns being violated. During this dissertation, different textual datasets having the symptoms and demographic information of covid patients were critically analysed, and during the analysis, it was found that long covid is more common in patients aged 60 and above. The probability of having long covid was same for the women as for the men. The most common symptoms that covid patients experienced were headache, body aches, cough, breathing difficulty, sore throat and fever. Then different ML models with different hyperparameters were trained on X-ray images of covid, normal and pneumonia patients to compare the classification efficacy of these models in classic and federated learning architectures. The same process was repeated for textual data, and it was found out that the accuracy, precision, and f-score dropped in federated architecture. Other than that, the training of models in federated architectures took more time as compared to the classical architectures.

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree, except where states otherwise by reference or acknowledgement, the work presented is entirely my own. If any act of plagiarism is found, I am fully responsible for every disciplinary action taken against me depending upon the seriousness of the proven offence.

Usman Shoukat

Acknowledgement

I like to thank my primary supervisor Pamela Thays Lins Bezerra, for her continuous support and guidance throughout the project. Her valuable insights and motivation lead the project to successful completion. I also like to thank my professor Angrosh Mandya for supporting me in the different areas of the project. I am also highly obliged to Rida Laraki for his insightful feedback during the different stages of the project. I also greatly acknowledge the assistance and learning opportunities provided by the University of Liverpool, especially by the Department of Computer Science, University of Liverpool.

I would also like to extend my sincerest gratitude to my family members, colleagues, and friends for their constant support, prayers, and constructive criticism throughout this project.

Contents

1. Introduction

Problem Description

Human life, health and productivity have largely been affected by the current covid-19 pandemic. During the start of the covid, the focus was on acutely ill people to save their lives and minimise the risks of the elderly and vulnerable. But now, a troubling phenomenon is coming to light with a staggering number of covid infected people saying that they are not recovering completely from covid symptoms over a longer period of time. This defines the long covid, which is not recovering from the symptoms which are being suggested as covid-19 symptoms for several weeks or months. [1] According to one study, the five most common symptoms found in long covid positive patients are fatigue, headache, dyspnoea, hoarse voice and myalgia. And the long covid was more common with increasing age and body mass index. [2]

As this long covid is affecting the lives of so many people, it is important to predict the long covid through the initial symptoms of covid so that it can be addressed earlier and in a better way by health practitioners.

Many different AI algorithms and models are recently being used to diagnose and predict the possibility of many high-risk diseases like breast cancer [3] and SARS-CoV-2 [4]. Thus, they can help us in the identification of COVID-19 as well [5]. Different AI algorithms and models can be trained to predict the possibility of long-term covid given the patient's health conditions and initial covid symptoms. But to train these traditional models, data collection is very critical. Medical data is usually not shared publicly to protect the patient's privacy concerns, which leads to insufficient datasets to train these models. [6] The non-availability of the datasets is limiting the number of research projects and solutions in the healthcare sector.

Here comes federated learning, a great tool to train a shared model by aggregating the locally computed updates, to rescue us. Google first introduced it in 2016. [7]

Every client (which can be a server, organisation, or mobile phone) has a local dataset and local machine learning model in federated learning. The global central server is in a federated environment that has a centralised learning model which aggregates the model parameters of all the local clients. Each client trains the local model by using its local data and then share the parameters of that locally trained model with the global model. The global model goes through many iterations to take the update from distributed clients. [8] As data never leaves the local centres in this type of learning thus, it is an excellent solution to the data privacy issues. The following figure demonstrates the federated learning

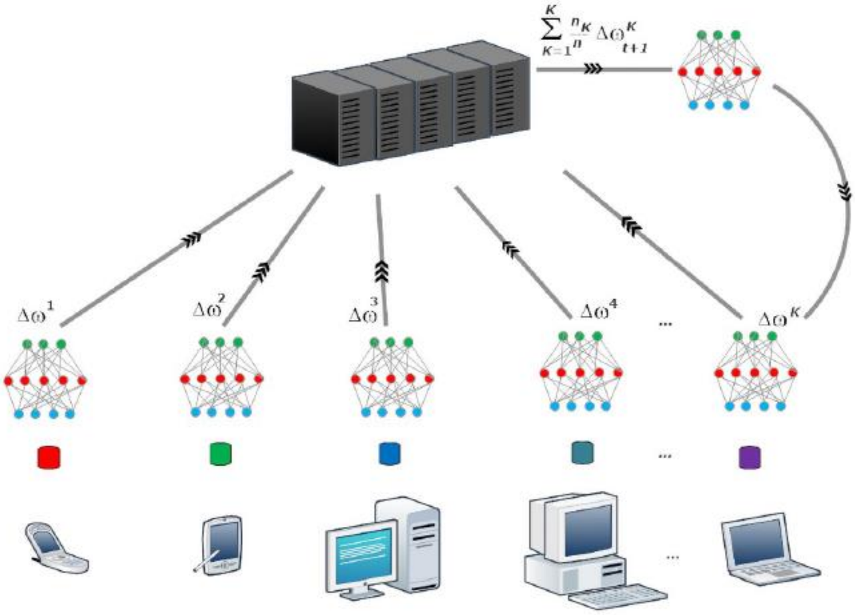


Figure 1 Federated Learning [8]

Aims and Objectives

The aim of the project is to study the application of federated AI for the diagnosis of long-term covid. The feasibility of the federated AI by comparing it with different classical machine learning algorithms was evaluated during the project. The main objectives of this project were:

- To research about the long covid, its symptoms and health condition that might lead to long covid

- To research and compare classical machine learning algorithms with federated learning methods. The comparison will be made on the same algorithms but between classic (or centralized) and federated learning architectures
- To learn about the API’s and tools like TensorFlow and Scikit learn needed to implement ML algorithms and federated learning
- To identify suitable datasets that contain the information about covid patients affected by long covid
- To train different machine learning models on the obtained datasets and compare their results with the federated learning architecture.

Initially, the project had to follow a methodology in which first a CNN model has to be built to classify between the Xray Images of the covid, normal and pneumonia patients. Then further SVM, Logistic Regression, KNN and MLP models were to be trained to classify between long-covid and short covid. And at the end, association rules on previous models were to be applied to give the final prediction on long-covid or short-covid.

A good dataset with covid symptoms and long and short covid labels could not be found or synthesized on which a classification model could be trained to classify between long-covid and short-covid during the project's progress. Therefore, the project scope was changed slightly, and most of the time was spent critically analysing the available datasets of covid patients. Unsupervised learning techniques are used to synthesize new long and short covid datasets during the project. Then CNN models are trained both in classical and federated architectures on the X-ray images dataset, and their results are compared. Furthermore, MLP models are trained both in classical and federated architectures on the Israeli government dataset (which have information about the demographic details and patients' symptoms), and their results are compared.

2. Background

Federated Learning Overview

Federated learning is one of the most emerging researches which are being studied in the fields of robotics, artificial intelligence and financial security. [9] It trains the decentralised statistical models over the remote or local datasets, e.g., data from mobile devices and hospitals, and then aggregates the trained parameters from these local devices to the global central model. [10]

The learning phase of the federated machine learning algorithms involves several communication rounds of global central server with the local client models. Initially, all the weights or the parameters of the models are initiallised with random values. Each communication round of the training process involves the following four steps:

Step 1:

First, the randomly initialized global model shares its parameters with the local models.

Step 2:

In the second step, each client or local model performs the training steps on the local data by minimizing the local objective function through epochs E.

Step 3:

When the training process is complete through enough Epochs, the local models share their model updates to the global server.

Step 4:

Finally, the central server receives the local model updates and updates the global model weights or parameters by averaging the received local model weights or parameters. It is done through the following equation:

$$W^t = \sum_{k=1}^{k} \frac{n^k}{n} W_k^t$$

Here t is the number of rounds, k is the number of clients, W^t are trained parameters or weights of the global model after round t, W_k^t are the weights of the client k after t communication rounds, n^k is the number of data points participated for client k, and n is the total number of data points participated in the training process. [11]

All the training process is shown in the following figure:

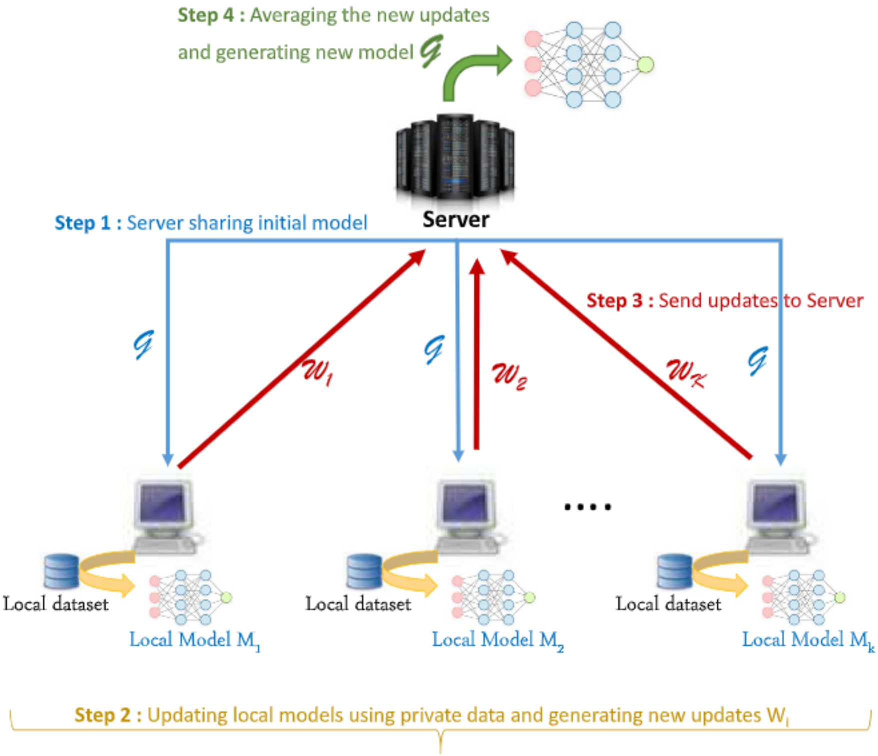

Step 4 : Averaging the new updates
and generating new model 𝒢
Server
Step 1 : Server sharing initial model
Step 3 : Send updates to Server
𝒢
𝒲1
𝒢
𝒲2
𝒲K
𝒢
....
Local dataset
Local Model M1
Local dataset
Local Model M2
Local dataset
Local Model Mk
Step 2 : Updating local models using private data and generating new updates Wi

Figure 2 Training Process of Federated Learning

Related Work

Recently, a lot of work is being done to develop deep learning algorithms such as a convolutional neural network to detect many diseases, e.g., detection of Covid-19 from the chest X-ray images of Covid and Normal patients.

In one of the studies [11], the authors experimented with dense convolutional neural networks by using the X-ray images of normal, pneumonia and covid patients. Models were trained both in centralized and federated learning architectures. VGG16 and ResNet50 architectures gave the best results among others by the use of normal and augmented data. Centralized models with normal data gave 93.75 % accuracy with

VGG16 model and 95.3 % with the ResNet50 model. The accuracy with the federated architecture on normal data was 93.75% for VGG16 and 95.4% for ResNet50.

In another study [12], the authors conducted X-ray images training experiments with MobileNet_v2, ResNet18, ResNeXt, and COVID-Net models. The sensitivity of these models on the testing set in the federated learning architecture was 86.83%, 91.26%, 90.37% and 89.17, respectively. After experiments, the study concluded that ResNet18 gave the best performance both in federated learning and classical architecture. ResNeXt had the better performance for the Covid-19 labelled images. MobileNet_v2 had the lowest number of trained parameters. The study concluded with ResNet18 and ResNeXt as the best models for the X-ray images of the Covid-19 dataset.

Another study [13] compares conventional federated learning with clustered federated learning. During the experimentation, VGG16 architecture with one extra convolutional layer and three extra fully connected dense layers was used. The model was trained with adam optimizer and a learning rate of 0.0001. The model was evaluated by calculated precision, recall and f-score. The precision of Federated learning, Federated learning (multi-modal), Clustered Federated Learning on covid class of the X-ray images dataset was 0.73, 0.3 and 0.71, respectively.

3. Ethical Use of Data

The university's ethical use of data policy was kept in mind while collecting the data from the online resources. The data needed for this project was about the demographic information of the covid patients, symptoms of covid experienced by these patients and X-ray images of the patients. All the data was collected from publicly available resources like government websites and kaggle; therefore, no permission was required from specific bodies to use this data.

4. Design

Datasets

Following datasets were used in the different stages of the project:

1. Chest X-ray (Covid-19 and pneumonia)
    - This data set contains the chest X-ray images of Covid-19, Pneumonia and normal patients [14]
    - This dataset was used for the classification into covid, pneumonia and normal patient
2. Israeli Government database
    - This repository had two datasets that were of interest. One contains the information about the covid symptoms experienced by the different patients, and the other dataset has information about the recovery period of different patients [15]
    - This dataset was analysed and used for the classification into positive and negative covid, both in classic and federated architectures, depending upon the symptoms and demographic information of the patient.
3. Prevalence of ongoing symptoms following coronavirus (COVID-19) infection in the UK
    - This dataset contains information about the self-reported long covid symptoms and the duration of these symptoms in the patients residing in the UK [16]
    - This data helped in identifying the most common symptoms experienced by covid patients
4. Preliminary dataset on confirmed cases of COVID-19, Public Health Agency of Canada
    - This dataset gives information about the asymptotic status, hospitalised status, death status, virus transmission status, age and gender of the patient [17]
    - This data was not useful as it was not giving any correlation between the symptoms and long and short covid.

5. Kaggle Dataset 1
    - This dataset has information about the symptoms of covid patients and the probability of having the covid infection [18]
6. Kaggle Dataset 2
    - This dataset has the information about the symptoms of covid patients, severity of these symptoms, and demographic information of the patient [19]

Programming Environment/Languages used

Different tutorials on Sci-kit Learn, Matplotlib, Pandas, Numpy [20], Tensorflow, Tensorflow Federated [21] [22] and PyTorch, aided to do the analysis on the textual and X-ray image datasets and to build the supervised and unsupervised learning models in Python 3. All the work was done in Python 3, and the following packages were used for the following specific purposes:

- Tensorflow for building classical CNN and MLP models
- Tensorflow-federated for building federated CNN and MLP models
- Pandas for data loading
- Sci-kit learn for building SVM, KNN, Logistic Regression, Perceptron and k-means models
- Seaborn and Matplotlib were used for plotting and visualising the data
- Microsoft Office Suite was used for documentation.

Machine Learning methods and Evaluation measures used

Supervised Learning Methods

Perceptron

Perceptron takes the motivation from the human nervous system. A neuron can learn only linear separable patterns and helps in binary classification. The model is trained by the external stimuli in the form of training data, and learning is done by changing the synaptic connections(weights) between the neuron and input data. The number of connections depends upon the number of features of input data. The input is passed through non-linear activation functions. Some examples of activation functions are unit, relu and softmax.

In unit function, the neuron is trained to fire on some specific threshold which means that the neuron fires if the activation score gained through the linear equation (a = WTX + b) is greater than that threshold θ, otherwise it does not fire. It is convenient to make the threshold equal to zero; for that purpose, we use bias and make it equal to - θ, which adjusts the threshold equal to zero. The following figure shows a neuron:

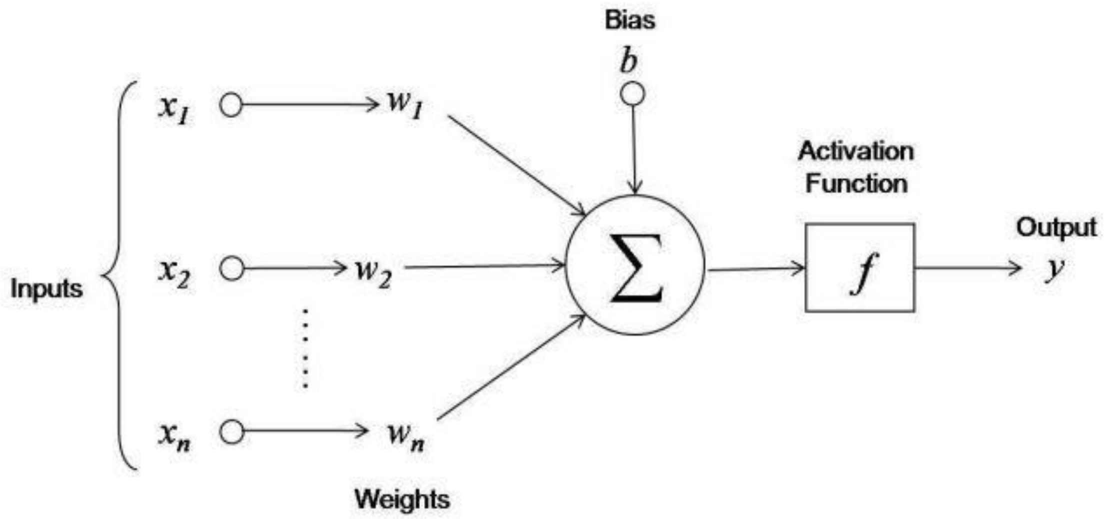


Figure 3 Neuron

Multi-Layer Perceptron

Perceptron can only do the linear binary classification. A complex network is needed fro a classification problem where data in non-linear and there are more than two classes to separate. This complex network includes one or more hidden layers and neurons between input and output layer. The output of the first layer is the input of the second layer, and so on. This complex network multiple hidden layers with multiple neurons is known as a multi-layer perceptron. The neurons and hidden layers in the multi-layer perceptron are hyperparameters and are set according to the complexity of data. The number of output neurons depends upon the number of classes in which we want to classify our data. [23] A sample neural network with two hidden layers is shown in the figure below.

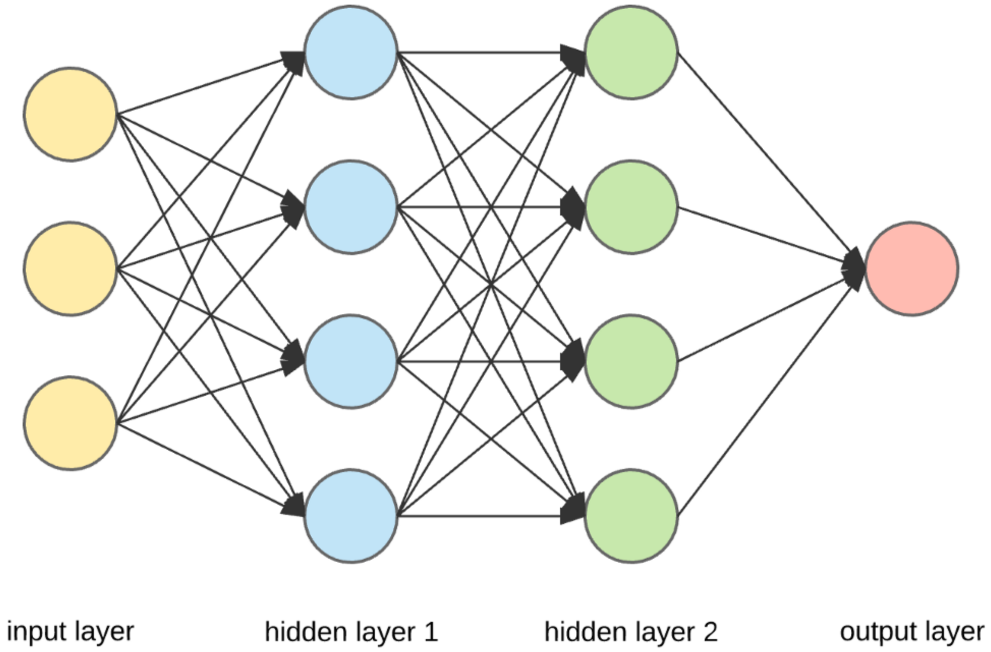


Figure 4 Multi-Layer Perceptron

Convolutional Neural Network(CNN)

Convolutional neural networks are among the most popular deep neural networks and are used for pattern recognition. The name of convolutional neural networks are after the linear mathematical operation of matrix called convolutions. CNN has multiple layers, including convolutional, non-linear, pooling, and fully connected. Convolutional and fully connected layers have parameters among these layers, but pooling and non-linear layers do not. Convolutional neural networks have excellent performance with image data classification problems, computer vision and natural language processing (NLP) [24]. The general architecture of convolutional neural networks is shown below:

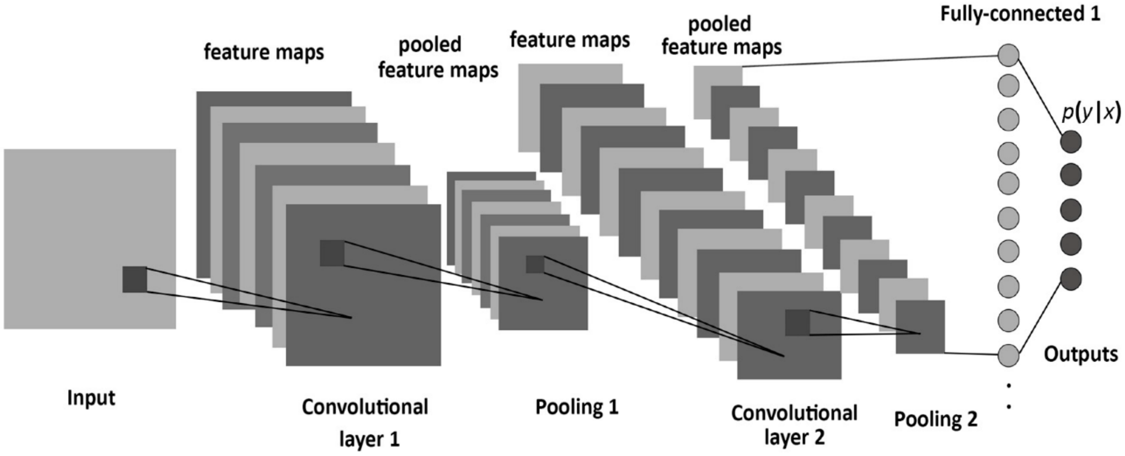


Figure 5 Convolutional Neural Network

Optimizers:

Optimizers are the functions or algorithms used to reduce the error of the cost function or increase the algorithm's efficiency. Some types of the optimisers are gradient descent, stochastic gradient descent, Adam, Adagrad and RMS-prop.

Gradient Descent:

Gradient Descent is the most widely used optimisation algorithm. It uses the first-order derivative of the loss function for the optimisation. It calculates the first-order derivative of the loss function and then decides to adjust the weights and biases of the neural network in such a direction that it reaches the minima. And then, after that, the loss is transferred from layer to layer through backpropagation. [25]

Stochastic Gradient Descent (SGD):

It is a variant of the gradient descent and updates the network's parameters more frequently than the gradient descent. In gradient descent, the model parameters are updated only once throughout the training process. Compared to that, the model parameters are updated after training through every data point in the dataset in SGD. [25]

Adagrad:

The above-mentioned optimisers have one common disadvantage: the learning rate remains constant for all the parameters. Adagrad is a variation that changes the learning rate for all the parameters at every time step t. It is a second-order optimization algorithm and works on optimizing the error function. [25]

Adam:

Adam (Adaptive moment estimation) is an optimizer that works with the momentum of first and second order. Momentum is introduced in the SGD optimizer to reduce the variance and accelerate the convergence in the right direction. While moving with momentum, we do not want to move with such a greater velocity that we pass over the minimum. Adam is the optimizer in which this control of velocity while moving with momentum is introduced. [25]

RMSProp:

RMSprop is the variation of the gradient descent optimizer with the momentum. The difference in RMSProp compared to gradient descent with momentum is that it also uses the adaptive learning rate. [26]

Logistic Regression

Logistic regression is the supervised machine learning algorithm that is used to predict the possibility of a target variable. This algorithm can only classify into two classes, and the dependant targetted variable is usually coded as 0 and 1.

Mathematically, logistic regression tells the probability of (Y=1) as a function of X and is used mostly for binary classification problems such as spam emails detection, cancer patients detection and diabetes prediction. [27]

Support Vector Machines

Support vector machine is the machine learning algorithm that is used for classification, regression and outlier detection problems. This algorithm can be used for the classification of two or more classes.

In a simple binary classification problem, the algorithm is trained to learn a straight line through the training set, which can separate the two different classes of data. Now, the data points towards one side of the line are labelled as class one, and the data points on the other side of the line are classified as class two. As there could be many lines drawn between the two types of classes, therefore to classify these classes efficiently and to draw a perfect line between the two classes, the margin of separation is used. The margin of separation is defined as the distance between the two decision planes and the nearest point d for each w and b (line parameters) [28]. A sample support vector machine is shown in the following figure:

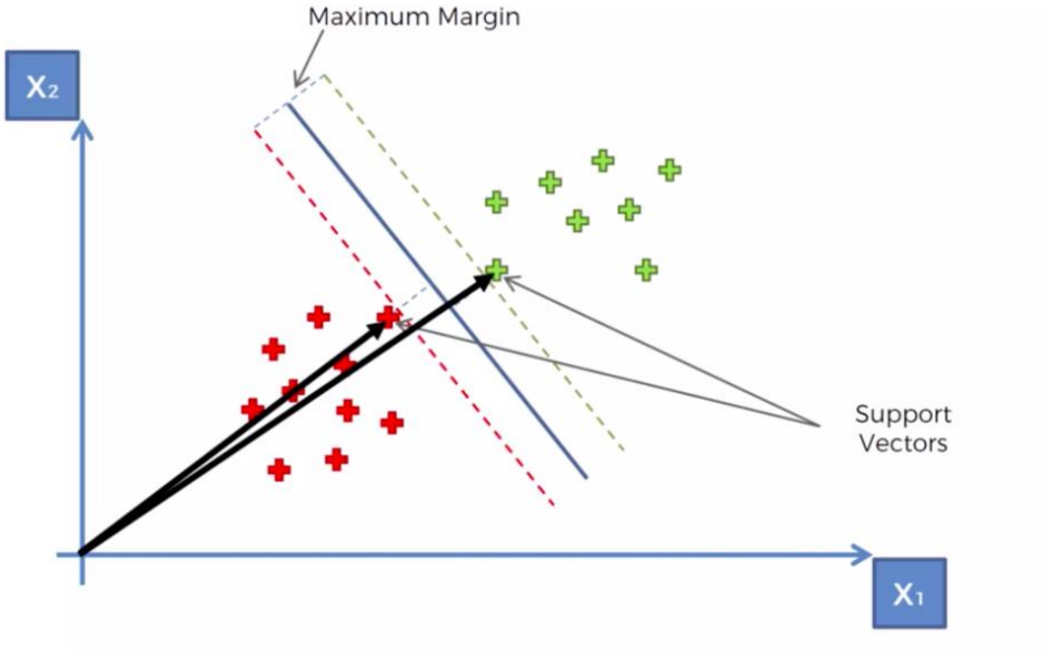


Figure 6 Support Vector Machine

To support the non-linearities in the data, different kernels, e.g., polynomial and radial basis functions, are used.

K-nearest Neighbours

KNN is one of the most fundamental and simple supervised machine learning models. It is usually the first priority when there is not enough knowledge about the data. This machine learning method is used for the classification problem.KNN is based on the euclidean distance between the test and training samples, with each sample in the training set labelled as one of the classes.

Before running the algorithm, a number of neighbours k is chosen. Then during the algorithm, the euclidean distance between the test sample and each of the training samples is calculated. In the end, closet k neighbours with the test sample are chosen depending upon the euclidean distance, and the class of the majority of the neighbouring samples is declared as the class of the test sample.

Unsupervised Learning Methods

K-Means Clustering

K-Means Clustering algorithm is one of the simple and most popular unsupervised machine learning algorithms. The objective of the k-means clustering algorithms is to group similar data points and discover their underlying patterns. For this purpose, the data is clustered in a k fixed number of clusters.

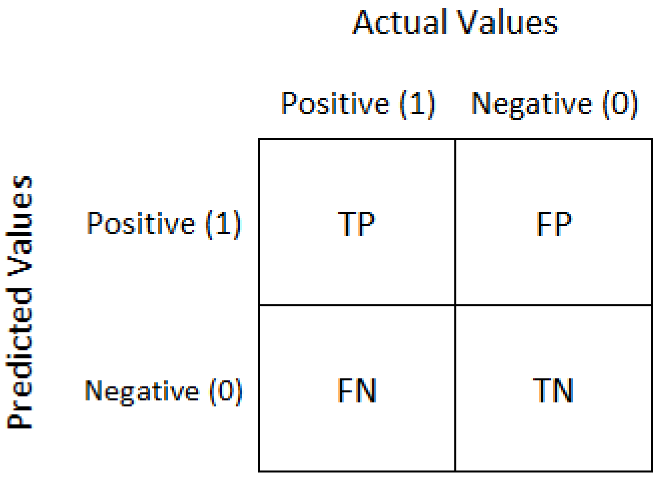
The algorithm starts with k centroids initialised randomly and represents each cluster. Then the distance of each data point is calculated across each of the k centroids. Depending upon that distance, the data point is assigned to the nearest cluster. When all the data points are assigned their clusters, then the means of each of the clusters is calculated and act as the new centroid.

The whole process is repeated unless the model converges or enough iterations have been reached. [29]

Evaluation measures

Confusion Matrix

The confusion matrix is an NxN matrix that is used for the evaluation of classification models. A sample 2x2 confusion matrix is shown in the following figure:



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Here, the target variable has only two classes, one being positive and the other being negative. The columns here represent the actual values, and the rows represent the predicted values.

There are four quadrants here in the matrix labelled as TP, TN, FP, FN.

- True positives (TP) are the predicted values same as the actual values, with both values being positive.
- True negatives (TN) are the predicted values the same as the actual values, with both values being negative.
- False positives (FP) are values falsely predicted as positives, but actually, they were negative
- False negatives (FN) are values falsely predicted as negatives, but actually, they were positive [30]

Precision

Precision tells the proportion of actual true positives among true predictions, and its formula is given by:

$$\frac{TP}{TP + FP}$$

Recall

Recall tells about the proportion of true positives that are correctly classified, and its formula is given by:

$$\frac{TP}{TP + FN}$$

Accuracy

Accuracy tells about the proportion data points that are correctly classified, and its formula is given by:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

F-Score

F-Score is the harmonic mean between precision and recall and is given by:

$$F - Score = \frac{2\ x\ precision\ x\ recall}{precision + recall}$$

Project Development Stages

The project was completed in five stages. In **the first stage**, the use of libraries like TensorFlow, Sci-kit Learn and PyTorch needed to implement the convolutional neural network was learned. Then classic convolutional neural networks were built to train on the x-ray images dataset [14], followed by the pre-processing of the x-ray images. These models can classify into three classes, i.e., Covid, Normal and Pneumonia patients.

During the **second stage** of the project, the available datasets having the information about covid symptoms and demographic information of the patients were critically analysed. K-means unsupervised learning technique was used to cluster the data with symptoms and demographic information into two clusters, i.e., long-covid and short-covid. After that, new data created after clustering was analysed, and machine learning algorithms like Support Vector Machines, Logistic regression, KNN and Perceptron were built on it. In the end, these models were evaluated by calculating the accuracy, precision and f-score.

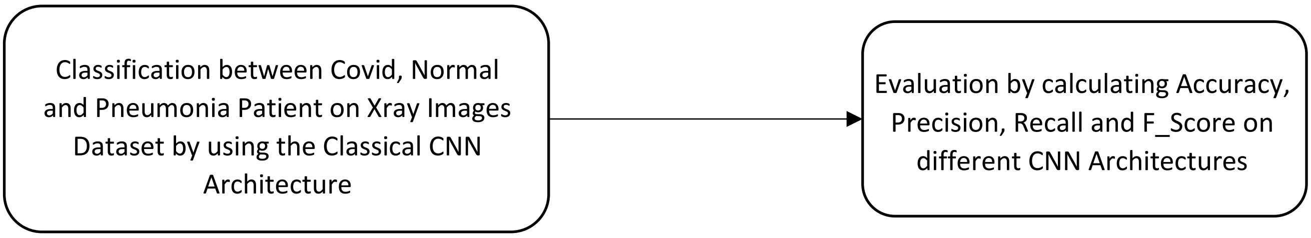
During the **third stage**, the CNN architectures with the best result in stage 1 were built in the federated environment and were evaluated by calculating accuracy, precision, and recall.

In **stage 4**, MLP models were trained in classic and federated learning architectures on Israeli textual dataset [15] and were evaluated by calculating the accuracy, precision and f-score.

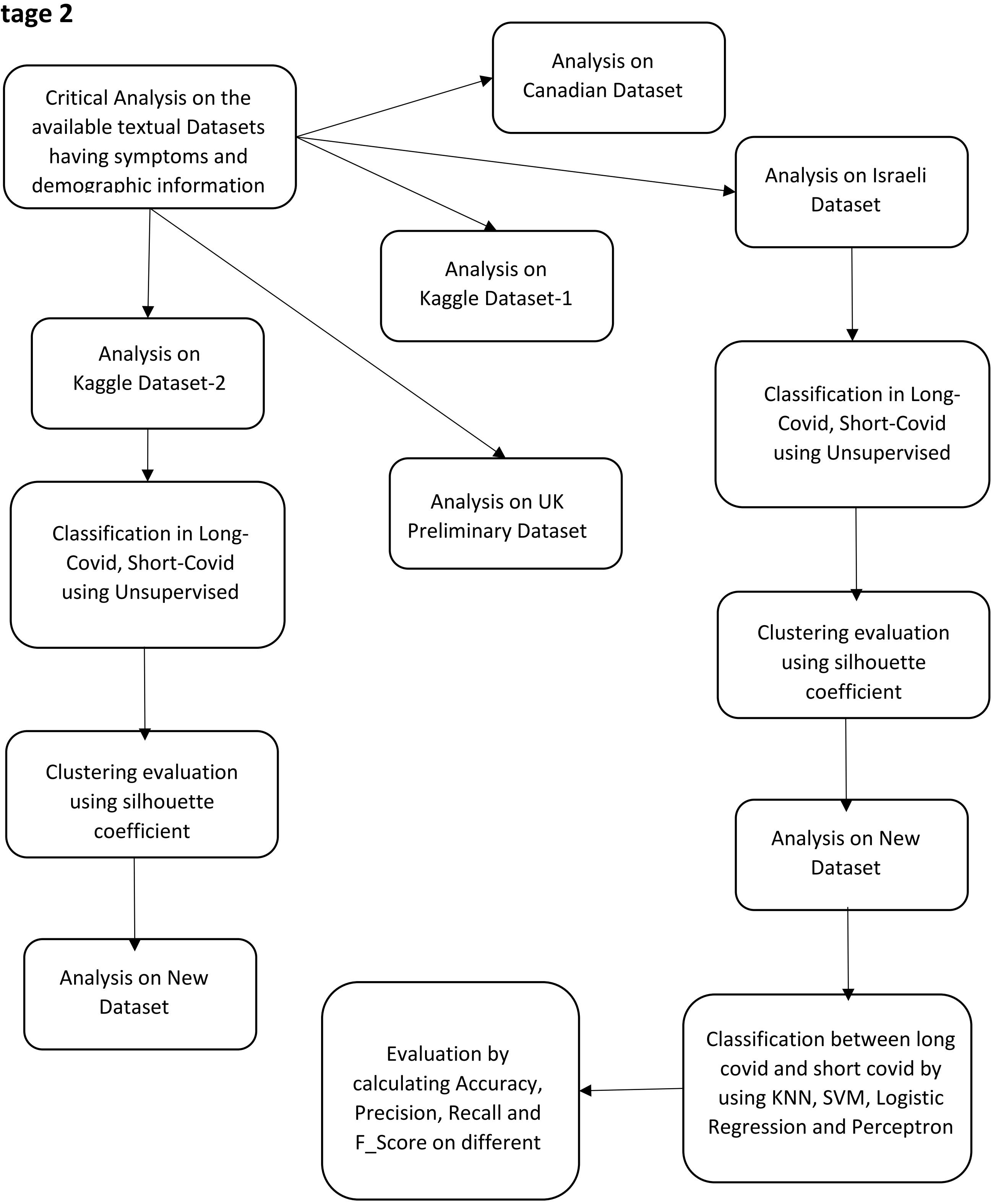
In **stage 5**, the results from the classic and federated architecture of CNN and MLPs were compared, and conclusions were drawn.

Following diagrams show the details of these stages:

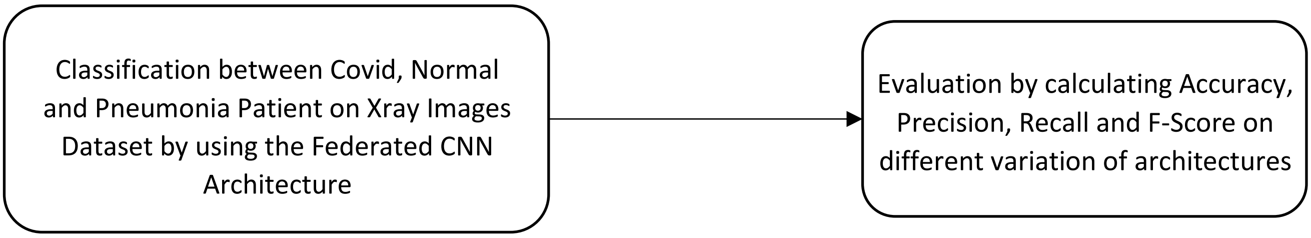
Stage 1


Classification between Covid, Normal and Pneumonia Patient on Xray Images Dataset by using the Classical CNN Architecture
Evaluation by calculating Accuracy, Precision, Recall and F_Score on different CNN Architectures

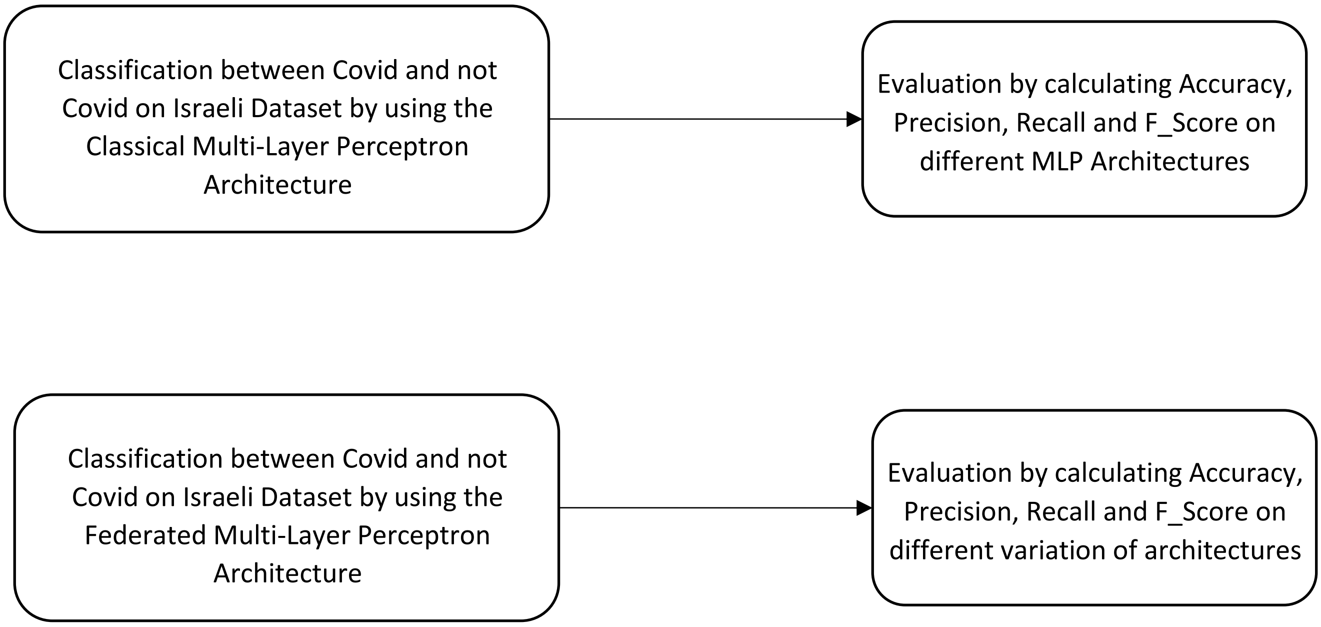
Stage 2


Critical Analysis on the available textual Datasets having symptoms and demographic information
Analysis on Canadian Dataset
Analysis on Israeli Dataset
Analysis on Kaggle Dataset-1
Analysis on Kaggle Dataset-2
Analysis on UK Preliminary Dataset
Classification in Long-Covid, Short-Covid using Unsupervised
Classification in Long-Covid, Short-Covid using Unsupervised
Clustering evaluation using silhouette coefficient
Clustering evaluation using silhouette coefficient
Analysis on New Dataset
Analysis on New Dataset
Classification between long covid and short covid by using KNN, SVM, Logistic Regression and Perceptron
Evaluation by calculating Accuracy, Precision, Recall and F_Score on different

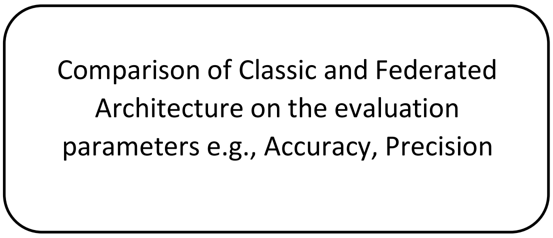
Stage 3


Classification between Covid, Normal and Pneumonia Patient on Xray Images Dataset by using the Federated CNN Architecture
Evaluation by calculating Accuracy, Precision, Recall and F-Score on different variation of architectures

Stage 4


Classification between Covid and not Covid on Israeli Dataset by using the Classical Multi-Layer Perceptron Architecture
Evaluation by calculating Accuracy, Precision, Recall and F_Score on different MLP Architectures
Classification between Covid and not Covid on Israeli Dataset by using the Federated Multi-Layer Perceptron Architecture
Evaluation by calculating Accuracy, Precision, Recall and F_Score on different variation of architectures

Stage 5


Comparison of Classic and Federated Architecture on the evaluation parameters e.g., Accuracy, Precision

5. Realisation

Stage 1

During this stage, different CNN models were trained on the X-ray images dataset [14], which can classify into Covid, Pneumonia and Normal patient. Different experiments were run with different CNN architectures by changing the hyperparameters and differeny optimizers.

Some of the hyperparameters that were changed during the experiments are as follows:

- Images size 130 pixels and 32 pixels
- Number of epochs 10, 15, 20
- Learning rate 0.001, 0.01
- Optimizers SGD, Adam
- Percentage of dropout 0.3, 0.5
- Different architectures with different combinations of Convolutional, Max pooling, Dense and dropout layers
- Different number of filters in Convolutional layer and different number of neurons in Dense layer

The X-ray dataset was huge, and each experiment took a long time. Other hyperparameters, e.g., filter size, padding, and pool size, could be experimented with. Due to shortage of time conventional filter size of 3x3 and pool size of 2x2 was picked during the experimentation. The activation function used in each layer of the convolutional network was “relu”. In the final dense layer “softmax” function was used to get the probability of each image belonging to each of the three classes.

The following results are from the three experiments which gave the best results:

Experiment No.	Architecture	Learning Rate	Optimizer	Epochs
1	Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Conv Layer(64 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Flattening Layer -> Dense Layer (32 neurons, activation “relu”) -> Dense Layer (64 neurons, activation “relu”) -> Dense Layer (3 neurons, activation “Softmax”)	0.01	Adam	10
2	Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Dense Layer (64 neurons, activation “relu”) -> Dense Layer (3 neurons, activation “Softmax”)	0.01	Adam	10
3	Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Flattening Layer -> Dense Layer (512 neurons, activation “relu”) -> Dropout Layer (Dropout percentage 0.3) -> Dense Layer (3 neurons, activation “Softmax”)	0.01	Adam	10

Results:

Experiment No.	Class	Precision	Recall	F-Score	Accuracy
1	COVID	0.91	0.95	0.93	0.91
	NORMAL	0.77	0.94	0.85	
	PNEUMONIA	0.98	0.89	0.93	
	Macro Avg	0.89	0.93	0.90	

	Weighted Avg	0.92	0.91	0.91	
2	COVID	0.91	0.79	0.85	0.87
	NORMAL	0.70	0.92	0.80	
	PNEUMONIA	0.96	0.86	0.91	
	Macro Avg	0.86	0.86	0.85	
	Weighted Avg	0.89	0.87	0.87	
3	COVID	0.98	0.83	0.90	0.91
	NORMAL	0.84	0.88	0.86	
	PNEUMONIA	0.94	0.94	0.94	
	Macro Avg	0.92	0.88	0.90	
	Weighted Avg	0.92	0.92	0.92	

Evaluation:

There was no significant change in the results with the experimentation of different hyperparameters. The accuracy was changed by 2-5 % during these experiments. Very simple architecture, e.g., One convolutional layer, one pooling layer, and one dense layer, gave a slightly lower accuracy because of underfitting. Highly complex architectures, e.g., five convolutional layers, four max-pooling layers, and four dense layers, also gave lower accuracy because of overfitting. The models with two convolutional layers, 1-2 max-pooling and 1-2 dense layers gave the best results.

Stage 2

This stage of the project consisted mostly of analysing the multiple textual datasets to obtain long and short term covid labelled datasets. Good datasets describing all symptoms were not readily available; therefore, this stage took longer than expected to complete. Different unsupervised learning algorithms were used to synthesize own dataset during this stage of the project.

Israeli Government database

This section critically analyses the Israeli Government dataset [15] with information about the patients' recovery period and demographic details. There were the following trends in the dataset:

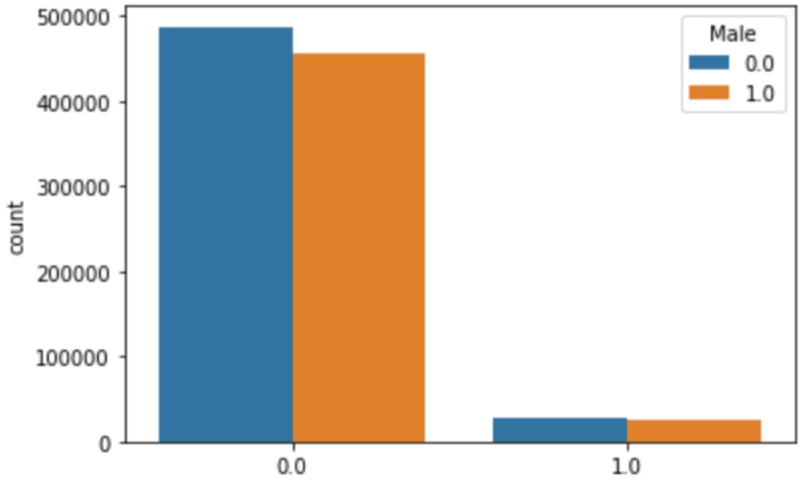


Figure 7 Long Covid vs Gender

In this graph, zero on the x-axis represent short covid, and 1 represents long covid. Similarly, the blue colour represents females, and the orange colour represents males.

The above graph clearly shows the number of patients having long covid is same in both the genders. Which means that long covid was independent of the gender of the patients and both the genders were equally affected by the long covid.

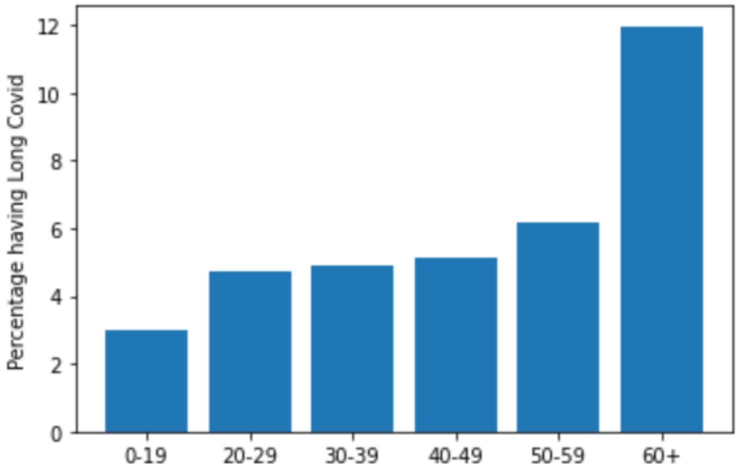


Figure 8 Age Group vs Covid

In the above graph the x-axis shows the age group, and the y-axis shows the percentage of patients with long-term symptoms among the covid positive tested patients. The above graph shows that the age group 60+ is more vulnerable to long-covid than other age groups. Almost 12% of 60+ patients had long term symptoms. The percentage of the patients in the age group, 0-19, having long term symptoms was relatively low, i.e., 3%. And 5.5% of the patients between 20 and 59 had the long covid.

Then, the other Israeli government Dataset, which had the information about symptoms of covid, was analysed. The following figure shows the correlation co-efficients of the features of the data:

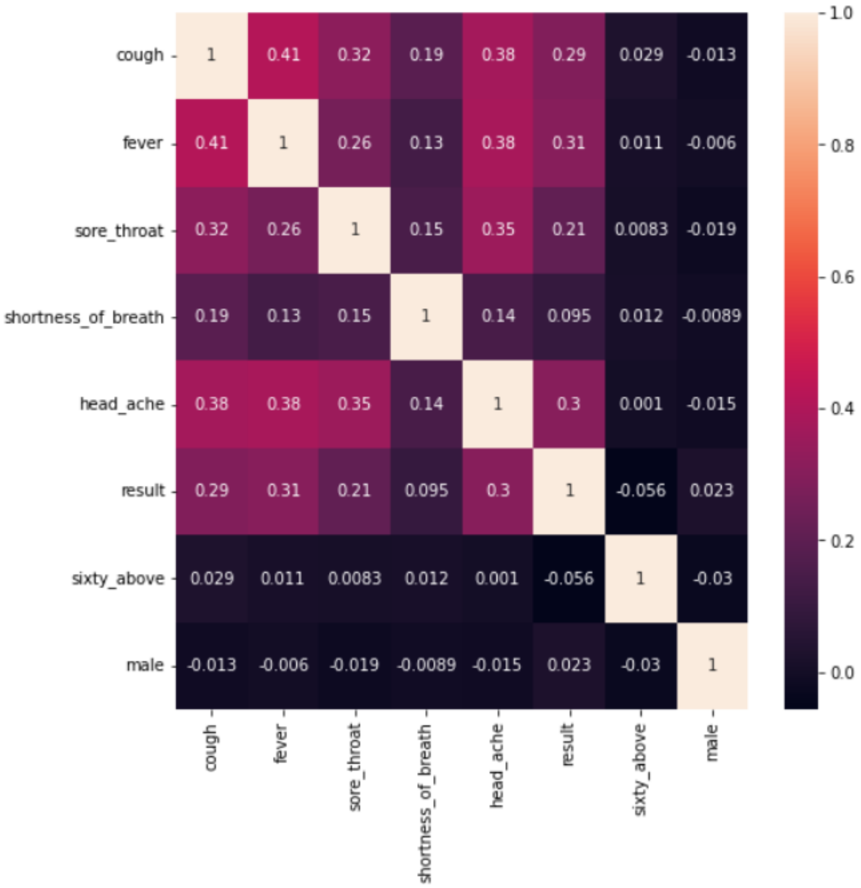


Figure 9 Correlation diagram of Israeli Dataset

On plotting the correlation matrix of the data, results showed that cough, fever and headache had a better correlation with the covid result, with correlation coefficient values being 0.29, 0.31 and 0.3, respectively. Although these correlation values are not very high and thus not showing strong correlation. One of the reasons for having the lower correlation coefficient values is the lower number of covid positive patients having these symptoms and most of the patients were asymptotic. But as mentioned above three symptoms had a greater value of correlation coefficient among others, so we can say cough, fever and headache are more prominent symptoms among others in the covid patients.

As this data was not giving any information about the long and short covid, therefore by using this Israeli dataset and unsupervised learning methods, a new dataset was synthesized. k-means clustering method was used to cluster the Israeli dataset. After clustering, the clusters were evaluated by using the silhouette coefficient.

Silhouette score with two clusters was 0.825 and silhouette score with three clusters was 0.55. As silhouette coefficient with two clusters had a higher value, therefore, the data was clustered in two clusters. On analysing the new created data, it was found that in cluster zero the proportion of people having the covid symptoms was far greater as compared to the cluster one as evident through the following graphs.

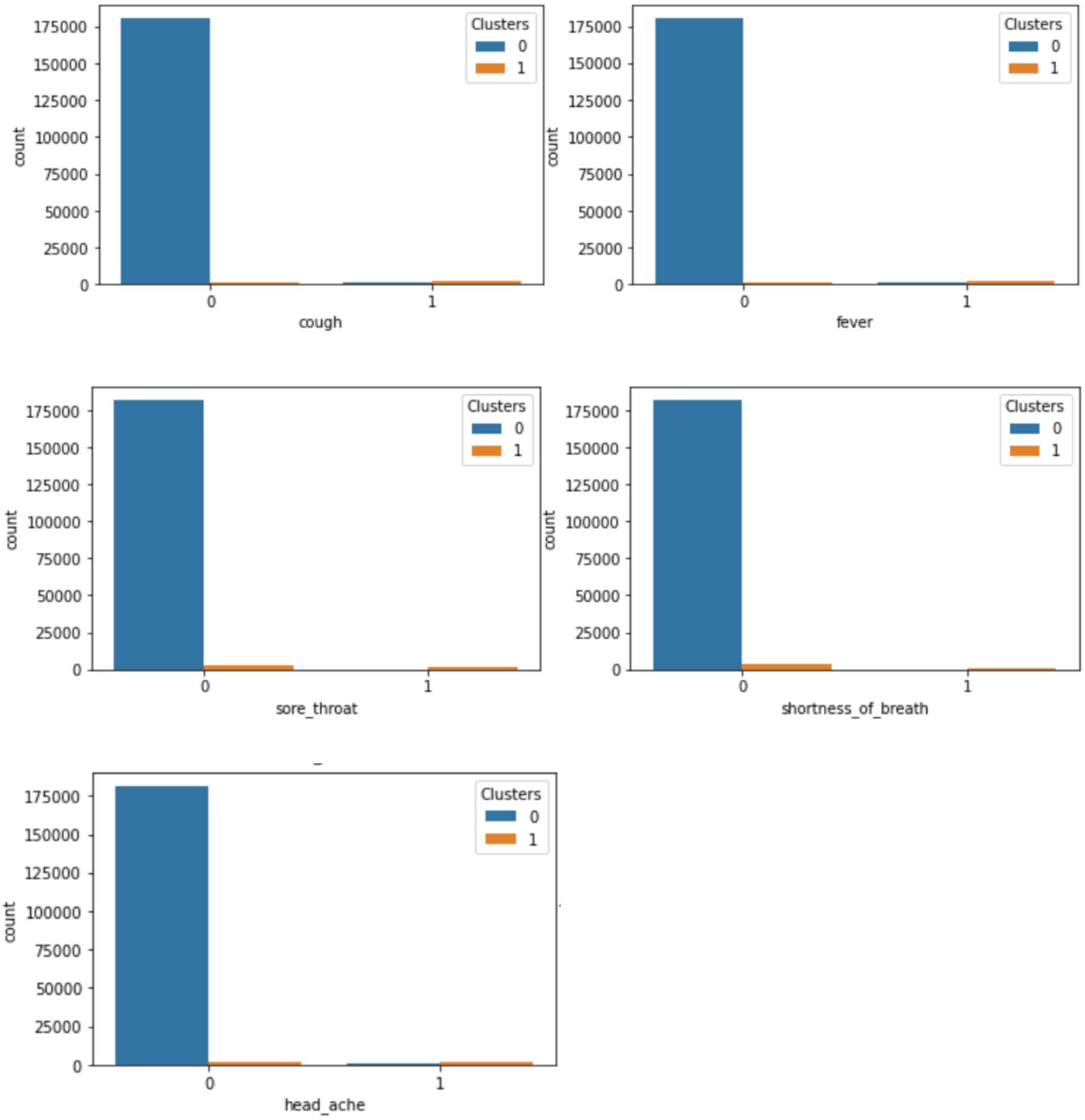


Figure 10 Symptoms vs Clusters

This pattern could also be seen in the correlation chart of the new created dataset. The “clusters” feature showed a high correlation coefficient value with the symptoms of covid. The correlation chart is shown below

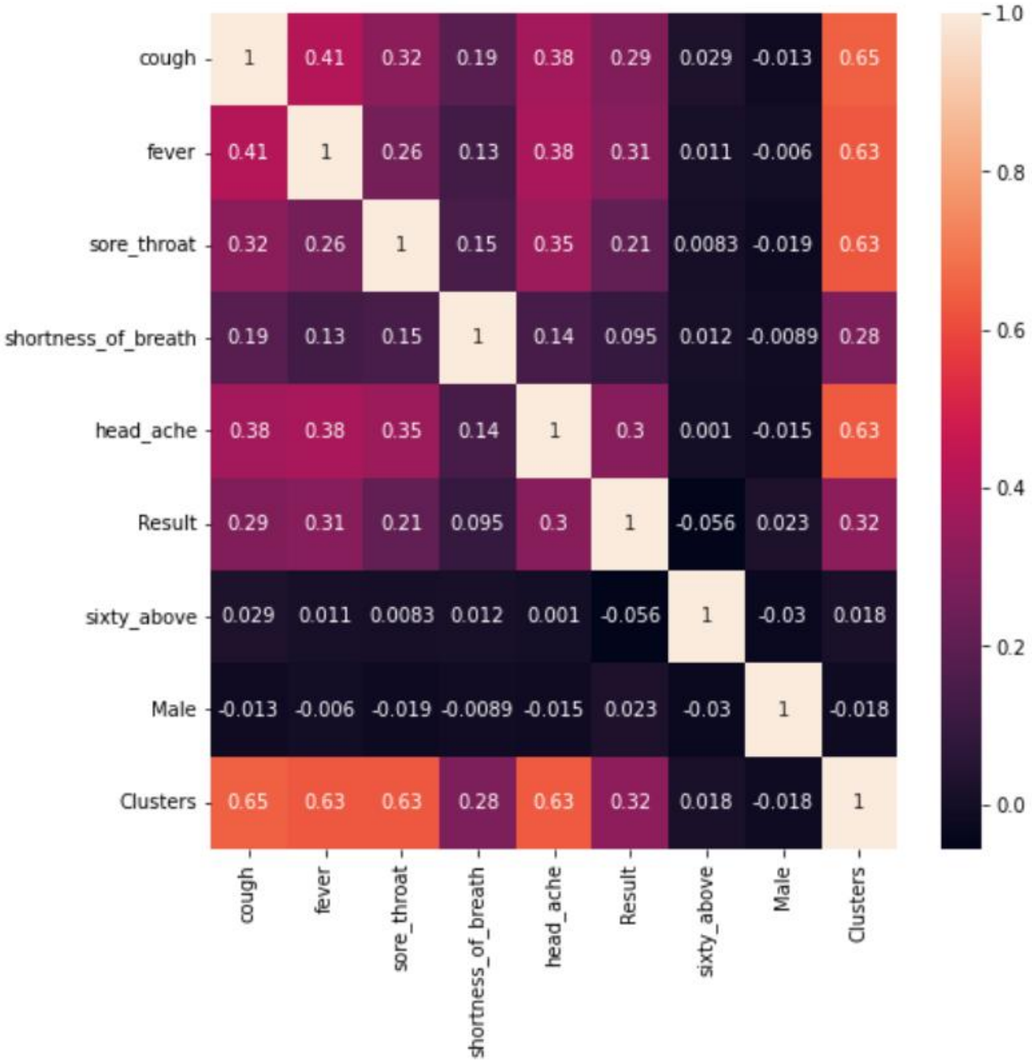


Figure 11 Correlation figure on Israeli Dataset after Clustering

As the conditions of patients in cluster one were severe compared to cluster zero, therefore cluster 1 was labelled as long covid and the cluster 0 as short covid.

Kaggle Dataset-1 and Kaggle Dataset-2

Kaggle Dataset-1 [19] was another dataset having information about symptoms of covid patients. This dataset was not showing good correlation between the features of dataset. Therefore, It was not further used. Following figure shows the correlation figure of that dataset.

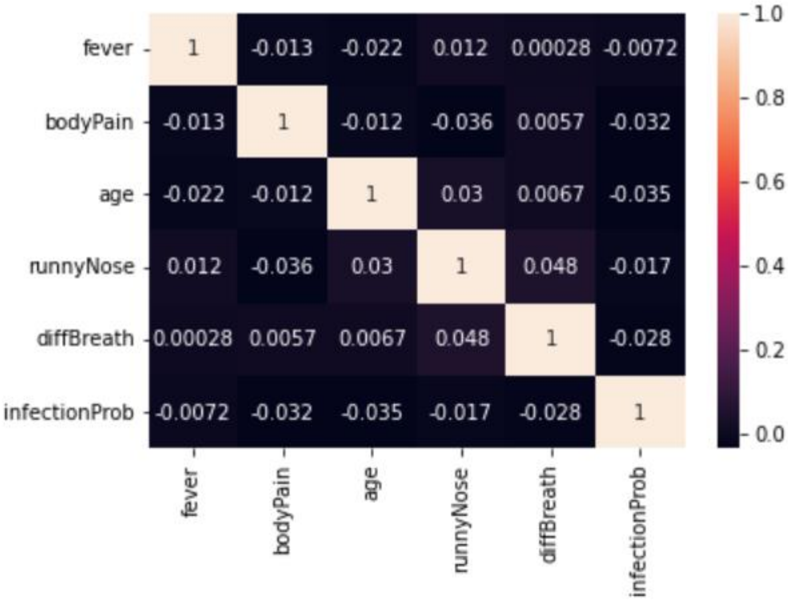


Figure 12 Correlation Kaggle Dataset-1

Then, the other kaggle dataset [31] that had information about covid patients' symptoms, demographic information, and information about the severity of the symptoms was analysed. Same process was followed for this dataset as was followed for the Israeli Government dataset. On analysing the new data created after clustering, It could be seen that there were almost equal number of patients with and without symptoms, and severity of these symptoms in each of the clusters. As the data was evenly spread out in terms of having the symptoms and severity of symptoms, therefore it was not a good dataset for the classification.

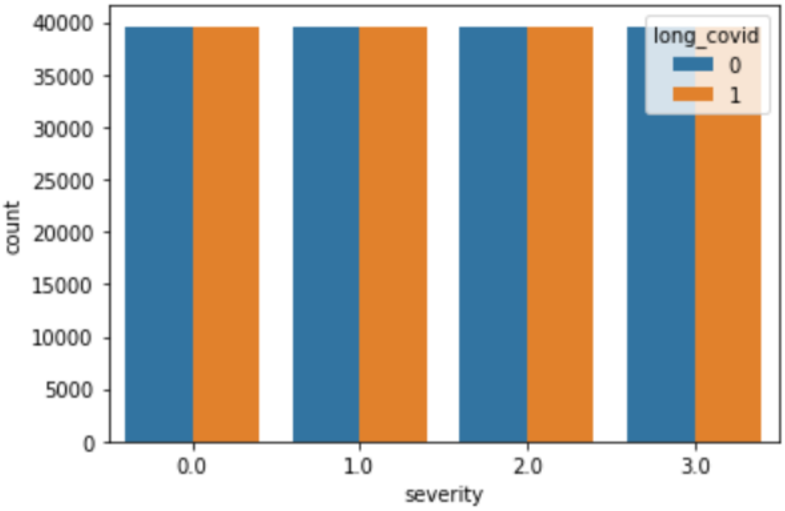


Figure 13 Severity Vs Clusters

In this plot, 0 shows no severity, 1 shows mild severity, 2 shows moderate severity and 3 shows high severity. Blue represents cluster 0 and orange represents cluster 1. This plot shows that there were equal number of people in both the clusters in every type of severity.

Classification

The new Israeli and Kaggle dataset obtained after clustering were used to build support vector machine, logistic regression, KNN and perceptron classifier models to classify the data into long-covid and short covid. Models were built with very basic architectures and the results were following for all these four different classifier models.

Results:

Model	Class	Precision	Recall	F-Score	Accuracy
Support Vector Machines	Covid Positive	1	1	1	1
	Covid Negative	1	1	1	
	Macro Avg	1	1	1	
	Weighted Avg	1	1	1	
Logistic Regression	Covid Positive	1	1	1	1
	Covid Negative	1	1	1	
	Macro Avg	1	1	1	

	Weighted Avg	1	1	1	
KNN with 2 Neighbours	Covid Positive	1	1	1	1
	Covid Negative	1	1	1	
	Macro Avg	1	1	1	
	Weighted Avg	1	1	1	
Perceptron	Covid Positive	1	1	1	1
	Covid Negative	1	1	1	
	Macro Avg	1	1	1	
	Weighted Avg	1	1	1	

Evaluation:

All these simple linear classifiers were able to classify the data perfectly in to labelled classes. On further analysis, it was found that all the features of the datasets were binary i.e., either 0 or 1. During the data clustering, the k-means algorithm had clustered them linearly. And because of this reason the classifiers were showing 100% accuracy and precision. As the dataset was now easily separable, therefore it was not a good data for drawing the comparison between classic and federated learning architectures on this dataset.

Canadian and UK dataset

The UK dataset [16] had statistical information, e.g., the information about the percentage of those who had a fever while being covid positive or the percentage of people who had a headache. For classification, the demographic information and the information of symptoms of each patient was needed. This dataset was not providing the needed information. Therefore, this dataset could not be used for the classification. On the other side, this dataset gave good insights into the most common symptoms covid patients were experiencing. The most common symptoms were cough, fever, sore throat, shortness of breath and headache.

The primary purpose of analysing these datasets was to find the correlations between symptoms, and long and short term covid. The Canadian dataset [17] was not giving any information about the symptoms patients experienced or the period of Covid infection. Therefore, this dataset was of no use for classification into long-covid and short covid.

Stage 3

Federated Learning Model on X-Ray Images

After completing the detailed analysis on the datasets available (previously explained), the development of the federated environment started. For doing this, it was necessary to install TensorFlow for federated learning and follow several tutorials. After having the knowledge to build federated architectures, models with best results in classic architecture were reimplemented in federated architecture. Several experiments were run with several different hyperparameters. Some of them are listed below:

- Images sizes 130 pixels and 32 pixels
- Number of epochs 10, 15, 20
- Learning rate 0.001, 0.01
- Client Optimizers SGD, Adam, Adagrad, RMSprop
- Server Optimizers SGD, Adam, Adagrad, RMSprop
- Number of clients 4, 5, 10
- Number of Rounds for training 10, 15, 20

During the experiments, one hyperparameter was changed in each experiment while keeping the other hyperparameters fixed. The results with some of the best combination of hyperparameters are as follows:

The following results are from the three experiments which gave the best results:

Experiment No.	Architecture	Learning Rate	Client Optimizer	Server Optimizer	Number of Clients	Number of Rounds	Epochs
1	Conv Layer(32 * 3x3 filters, activation "relu") -> Max pooling layer (2x2) -> Conv	0.01	Adagrad	RMSprop	4	15	10

	Layer(64 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Flattening Layer -> Dense Layer (32 neurons, activation “relu”) -> Dense Layer (64 neurons, activation “relu”) -> Dense Layer (3 neurons, activation “Softmax”)						
2	Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Dense Layer (64 neurons, activation “relu”) -> Dense Layer (3 neurons, activation “Softmax”)	0.01	Adagrad	RMSprop	5	15	10
3	Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Conv Layer(32 * 3x3 filters, activation “relu”) -> Max pooling layer (2x2) -> Flattening Layer -> Dense Layer (512 neurons, activation “relu”) -> Dropout Layer (Dropout percentage 0.3) -> Dense Layer (3 neurons, activation “Softmax”)	0.01	Adagrad	RMSprop	10	15	10

Results:

Experiment No.	Class	Precision	Recall	F-Score	Accuracy
1	COVID	0.95	0.47	0.63	0.93
	NORMAL	0.67	0.88	0.76	
	PNEUMONIA	0.91	0.86	0.88	
	Macro Avg	0.84	0.74	0.76	
	Weighted Avg	0.85	0.83	0.83	
2	COVID	0.94	0.73	0.83	0.85
	NORMAL	0.82	0.71	0.70	
	PNEUMONIA	0.85	0.96	0.90	
	Macro Avg	0.87	0.77	0.81	
	Weighted Avg	0.85	0.85	0.84	
3	COVID	0.82	0.91	0.86	0.88
	NORMAL	0.79	0.79	0.79	
	PNEUMONIA	0.92	0.91	0.92	
	Macro Avg	0.84	0.87	0.86	
	Weighted Avg	0.88	0.88	0.88	

During the experiments, Adam and SGD were not giving good results. Comparatively, higher accuracy and precision was achieved with Adagrad and RMSprop. Furthermore, the accuracy was dropping in most of the experiments with an increasing number of clients. The reason behind that was the same amount of data was distributed now among more clients, and thus, each client had a lower amount of data to learn the pattern.

Stage 4

The purpose of this section of the project was to compare the results from classical architectures with the federated architectures on the textual dataset of the covid patients. Given that it was not possible to synthesize good long and short term covid-19 datasets, the use of Multi-Layer Perceptron and other machine learning algorithms in classical and federated architectures was analysed on the covid positive and covid negative classification.

Classical Machine Learning algorithms on Israeli Dataset

Different experiments were done with different machine learning algorithms which includes Support Vector machines, Logistic Regression model, KNN model and Perceptron to classify the Israeli Dataset in covid positive and covid negative. The results from some of these experiments were following:

Model	Class	Precision	Recall	F-Score	Accuracy
Support Vector Machines	Covid Negative	0.90	0.99	0.94	0.90
	Covid Positive	0.84	0.25	0.39	
	Macro Avg	0.87	0.62	0.67	
	Weighted Avg	0.89	0.90	0.87	
Logistic Regression	Covid Negative	0.90	0.99	0.94	0.89
	Covid Positive	0.84	0.23	0.36	
	Macro Avg	0.87	0.61	0.65	
	Weighted Avg	0.89	0.89	0.87	
KNN with 2 Neighbours	Covid Negative	0.90	0.99	0.94	0.89
	Covid Positive	0.85	0.22	0.34	
	Macro Avg	0.87	0.60	0.64	
	Weighted Avg	0.89	0.89	0.87	
Perceptron	Covid Negative	0.89	1	0.94	0.89
	Covid Positive	0.84	0.16	0.27	
	Macro Avg	0.87	0.58	0.61	
	Weighted Avg	0.88	0.89	0.85	

The basic version of all the above mentioned machine learning classification algorithms gave almost the same results with 89 % accuracy.

Then I experimented with different multi layer perceptron models and different hyperparameters e.g., different architecture, different number of epochs, different optimizers and different learning rates to increase the complexity of the model. During these experiments, one of the above mentioned hyperparameters were changed while fixing the others. One of the models giving good results was:

Model:

```
_________________________________________________________________
                           Layer (type)
=================================================================
     Dense layer(64 hidden neurons, activation function “relu”)
_________________________________________________________________ D
     ense layer(128 hidden neurons, activation function “relu”)
 _________________________________________________________________
    Dense layer(2 hidden neurons, activation function “softmax”)
=================================================================
Learning rate 0.01, optimizer Adam, epochs 10
```

Results

```
              precision    recall  f1-score   support

    NEGATIVE       0.90      0.99      0.94     16176
    POSITIVE       0.84      0.27      0.41      2433

    accuracy                           0.90     18609
   macro avg       0.87      0.63      0.68     18609
weighted avg       0.89      0.90      0.87     18609
```

Which was same as for the other machine learning algorithms.

The accuracy of these models was good, but the recall and f-score were very low on the positive class. The reason behind getting the lower recall and f-score was the uneven data of both the classes. In the dataset, there were 24,000 instances of positive class and 162,000 instances of the negative class. Because of that, the model was getting biased towards the negative class.

I further tried to optimise the results by reducing the instances of negative class to same as the same number of instances of positive class. After training the MLP on the new reduced dataset, the results were following:

Model:

```
_____________________________________________________________
                        Layer (type)
=============================================================
     Dense layer(64 hidden neurons, activation function "relu")
 _____________________________________________________________
    Dense layer(512 hidden neurons, activation function "relu")
 _____________________________________________________________
          Dropout layer with probability of dropping 0.5
 _____________________________________________________________
   Dense layer(2 hidden neurons, activation function "softmax")
=============================================================
           Learning rate 0.01, optimizer Adam, epochs 10
```

Results:

```
              precision    recall  f1-score   support

    Negative       0.74      1.00      0.85     10054
    Positive       0.97      0.26      0.41      4751

    accuracy                           0.76     14805
   macro avg       0.85      0.63      0.63     14805
weighted avg       0.81      0.76      0.71     14805
```

This reduction in data increased the precision on positive class but at the cost of overall accuracy. The precision on positive class jumped to 97 % and accuracy dropped to 76 %.

Federated MLP Classification on Israeli Dataset

After training the MLP model in classical architecture, the MLP model was trained in the federated environment. Different experiments were run with different federated learning hyperparameters and the same models that gave the best results in classical architecture. Different experiments with the different number of rounds, different client and server optimizers, and different number of clients were run. The results with of the best combination of hyperparameters were following:

Model:

```
_________________________________________________________________
                           Layer (type)
=================================================================
    Dense layer(64 hidden neurons, activation function "relu")
_________________________________________________________________ D
    ense layer(128 hidden neurons, activation function "relu")
 _________________________________________________________________
   Dense layer(2 hidden neurons, activation function "softmax")
=================================================================
          Learning rate 0.001, client optimizer Adagrad,

     server optimizer RMSprop, epochs 10, Number of Clients 5

                        Number of rounds 10
```

Results

```
              precision    recall  f1-score   support

    NEGATIVE       0.90      0.99      0.94     16176
    POSITIVE       0.84      0.23      0.36      2433

    accuracy                           0.89     18609
   macro avg       0.87      0.61      0.65     18609
weighted avg       0.89      0.89      0.87     18609
```

Federated architecture gave almost the same results as classical architecture because of enough data evenly distributed among the clients to train the model.

Stage 5

Comparison of Classical and Federated Learning Architectures

On comparing the results I got from classical and federated architectures, I could see that in federated environment, accuracy, precision, recall, and f-score were less compared to classical architectures. I am attaching here the results of architecture that gave me the best results.

Model

```
_________________________________________________________________
                          Layer (type)
=================================================================
Convolutional layer(32 filters of size 3x3, activation function “re
                              lu”)
_________________________________________________________________
Convolutional layer(32 filters of size 3x3, activation function “re
                              lu”)
_________________________________________________________________
                    Max pooling layer of 2x2
_________________________________________________________________
_________________________________________________________________
                       Flattening Layer
_________________________________________________________________
    Dense layer(512 hidden neurons, activation function “relu”)
 _________________________________________________________________
           Dropout layer with dropout percentage of 0.3
_________________________________________________________________
   Dense layer(3 hidden neurons, activation function “softmax”)
=================================================================
          Learning rate 0.001, clinet optimizer Adagrad,

    server optimizer RMSprop, epochs 10, Number of Clinets 10,

                       Number of rounds 15
```

Classical architecture results:

```
              precision    recall  f1-score   support

       COVID       0.98      0.83      0.90       116
      NORMAL       0.84      0.88      0.86       317
   PNEUMONIA       0.94      0.94      0.94       855

    accuracy                           0.92      1288
   macro avg       0.92      0.88      0.90      1288
weighted avg       0.92      0.92      0.92      1288
```

Federated Architecture Results:

Result

```
              precision    recall  f1-score   support

       COVID       0.82      0.91      0.86       116
      NORMAL       0.79      0.79      0.79       317
   PNEUMONIA       0.92      0.91      0.92       855

    accuracy                           0.88      1288
   macro avg       0.84      0.87      0.86      1288
weighted avg       0.88      0.88      0.88      1288
```

In the federated environment for X-ray classification, the accuracy dropped by 4%, precision by 16% and f-score by 4%. The recall of COVID class increased in federated architecture by 8% as compared to classical architecture.

The reason for having the accuracy and precision drop was that each local model was now being trained on lower number of datapoints. This precision, accuracy and f-score can be increased by introducing more datapoints to the dataset.

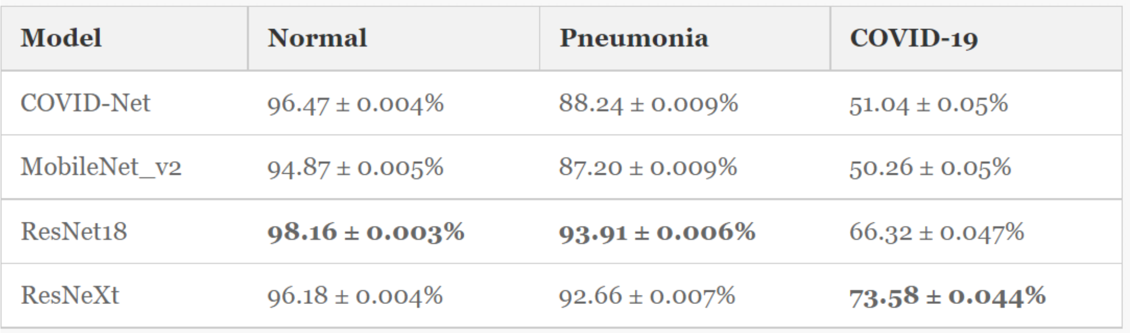
In the textual Israeli dataset, the results were almost similar both in classic and federated architectures because of the reason that each local model of the client had enough data to learn the pattern in the dataset.

6. Comparison with related work

In one of the studies on the similar data of X-ray images, many different complex architectures were trained to classify the X-ray images into covid, pneumonia and normal patient X-ray. VGG16 and ResNet50 models were giving very good results and the results were as follows: [32]

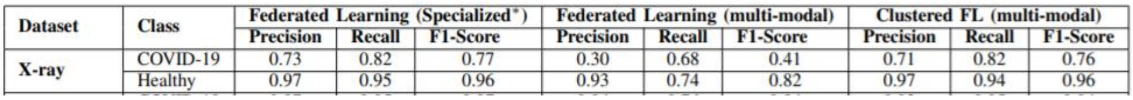
Method	Accuracy
FL-VGG16	93.57%
FL-VGG16 + data augmentation	94.4%
Centralized-VGG16	93.75%
Centralized-VGG16 + data augmentation	94%
FL-ResNet50	95.4%
FL-ResNet50 + data augmentation	97%
Centralized-ResNet50	95.3%
Centralized-ResNet50 + data augmentation	96.5%

Another study [12] that trains and draws comparison among MobileNet_v2, ResNet18, ResNeXt, and COVID-Net models calculated the perplexity of these models on each class of the dataset. The results were following:



Model	Normal	Pneumonia	COVID-19
COVID-Net	96.47 ± 0.004%	88.24 ± 0.009%	51.04 ± 0.05%
MobileNet_v2	94.87 ± 0.005%	87.20 ± 0.009%	50.26 ± 0.05%
ResNet18	**98.16 ± 0.003%**	**93.91 ± 0.006%**	66.32 ± 0.047%
ResNeXt	96.18 ± 0.004%	92.66 ± 0.007%	**73.58 ± 0.044%**

In another study [13], the comparison of conventional federated learning with the clustered federated learning on VGG16 architecture with one extra convolutional and three extra dense layers gave the following results:



Dataset	Class	Federated Learning (Specialized*)			Federated Learning (multi-modal)			Clustered FL (multi-modal)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
X-ray	COVID-19	0.73	0.82	0.77	0.30	0.68	0.41	0.71	0.82	0.76
	Healthy	0.97	0.95	0.96	0.93	0.74	0.82	0.97	0.94	0.96

As compared to these results, the results of my best models were following:

Method	Class	Precision	Recall	F-Score	Accuracy
CNN-Classical	COVID	0.98	0.83	0.90	92%
	NORMAL	0.84	0.88	0.86	
	PNEUMONIA	0.94	0.94	0.94	
	Macro Avg	0.92	0.88	0.90	
	Weighted Avg	0.92	0.92	0.92	
CNN-Federated Learning	COVID	0.82	0.91	0.86	88%
	NORMAL	0.79	0.79	0.79	
	PNEUMONIA	0.92	0.91	0.92	
	Macro Avg	0.84	0.87	0.86	
	Weighted Avg	0.88	0.88	0.88	

7. Learning Points

A breif overview of the things that I learned during the project are time management skills, writing and presentation skills, and research skills.

During the project, I learned about the practical implementation of machine learning algorithms on real word problem. Although, course curriculum gave me good theoretical understanding of the basics and mathematical background of different machine learning algorithms and their evaluation techniques but implementing them practically gave me confidence on the grasp on those theoretical concepts.

During the project, I learned about new libraries that were needed for the success of the project which includes Tensorflow, Tensorflow federated, Scikit Learn and PyTorch. This project gave me exposure of handling, visualising and analysing data by the use of libraries including Numpy, Seaborn and Matplotlib.

During the installation of libraries, different dependency conflict was raising. This gave me a good experience of using the online resources to tackle such problems. Getting help from online resources while resolving different type of errors in the code was a great learning experience.

Federated learning was not included in my course curriculum. Learning and researching a new topic on my own was a great learning opportunity for me. It gave me confidence to research about and explore many other different areas of the project as well. This project has really inculcated in me the passion and drive to research.

In terms of technical aspects, the project helped me nurturing my programming skills as well as writing skills while documenting my work.

8. Conclusion

During the analysis of different covid datasets, it was found that long covid is equally affecting both the genders i.e., male and female. The people with higher age group are more prone to having long covid and this trend is highly prominent with age group greater than 60.

During the experimentation and comparison of results from classical and federated architectures, it was inferred that the accuracy, precision and f-score dropped for the X-ray images dataset dropped in federated architecture because less data was now assigned to each local model. Which was indeed resolving the privacy issues but at the cost of 4% accuracy, 16% precision and 4% f-score.

On the other hand, the evaluation parameters of the textual dataset remained same in both type of architectures because of the spreading out of enough data among the clients that could help the local models to learn the underlying patterns in the data sent locally to each client.

During the experimentation, the models trained in the federated architectures were taking longer to train as compared to the classical architectures.

9. Challenges

During the project, I experienced the following challenges:

- Getting the right dataset for the classification of long-covid and short covid
- The X-ray images data was huge, and it was taking a longer time to train each model. Therefore, experimenting with different architectures and hyperparameters both in classical and federated environments was challenging
- As federated learning is a relatively new topic, therefore very few tutorials were available on federated learning and TensorFlow federated. On top of that, the available tutorials were on different versions of TensorFlow federated; consequently, it was a bit difficult to figure out which functions were available in which versions
- Different versions of TensorFlow were raising dependency conflicts. It was challenging to resolve all the dependency issues every time I was installing a new library.

10. Future Work

Because of the scarcity of the long and short data, it was difficult to train classification algorithms that could classify between the long and short efficiently. As part of the future work, good datasets can be synthesized or created which can efficiently train the machine learning algorithms.

The performance of machine learning models on X-ray images was dropping while training in the federated architecture. This was because of less data available for distributing among local clients. Next time X-ray images from different resources can be merged to increase the database or new data can be generated by the use of data augmentation techniques.

The overall performance of machine learning algorithms both in classical as well as federated architectures can be improved by the use of transfer learning of complex architectures like VGG16 and ResNet50 which have been developed in the [32] study.

11. Appendix

All the codes used for this project are available at the following GitHub page: https://github.com/usmanshouk/MSc-Project

12. References

[1] N. Nabavi, “Long covid: How to define it and how to manage it.,” 2020.

[2] C. Sudre, B. Murray, T. Varsavsky, M. Graham, R. Penfold, R. Bowyer, J. Pujol, K. Klaser, M. Antonelli, L. Canas and E. Molteni, “Attributes and predictors of long COVID,” pp. 626-631, 2021.

[3] K. Malherbe, “Tumor microenvironment and the role of artificial intelligence in breast cancer detection and prognosis,” *The American journal of pathology,* 2021.

[4] C. D'Ambrosia, C. Henrik and A.-S. Eliah, “Computing SARS-CoV-2 infection risk from symptoms, imaging, and test data: Diagnostic model development.,” *Journal of Medical Internet Research,* vol. 12, 2020.

[5] B. Liu, B. Yan , Y. Zhou , Y. Yang and Y. Zhang, “Experiments of federated learning for covid-19 chest x-ray images,” *arXiv preprint arXiv:2007.05592,* 2020.

[6] W. Zhang, T. Zhou, Q. Lu, X. Wang, C. Zhu, H. Sun, Z. Wang, S. Lo and F. Wang, “Dynamic fusion-based federated learning for COVID-19 detection,” *IEEE Internet of Things Journal,* 2021.

[7] McMaham, Brendan, M. Eider , R. Daniel , H. Seth and y. A. Blaise Aguera , “Communication-efficient learning of deep networks from decentralized data,” *Artificial intelligence and statistics,* no. PMLR, pp. 1273-1282, 2017.

[8] S. Abdul , Mustafa, T. Sanaa and R. Mohamed, “COVID-19 detection using federated machine learning,” no. Plos one, 16(6), e0252573, 2021.

[9] B. Liu, B. Yan, Y. Zhou, Y. Yang and y. Zhang, “Experiments of federated learning for covid-19 chest x-ray images,” 2020.

[10] T. Li, A. Sahu, A. Talwalkar and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine,* pp. 50-60, 2020.

[11] I. Feki, S. Ammar, Y. Kessentini and K. Muhammad, “Federated learning for COVID-19 screening from Chest X-ray images,” *Applied Soft Computing,* 2021.

[12] B. Yan , J. Wang, J. Cheng, Y. Zhou, Y. Zhang, Y. Yang, L. Liu, H. Zhao, C. Wang and B. Liu , “Experiments of federated learning for covid-19 chest x-ray image,” *International Conference on Artificial Intelligence and Security ,* July 2021.

[13] A. Qayyum, K. Ahmad, M. Ahsan, A. Al-Fuqaha and J. Qadir, “Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge.,” *Jan 2021.*

[14] P. Patel, “Chest X-ray (Covid-19 & Pneumonia),” 2020. [Online]. Available: https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia.

[15] “COVID-19 reservoir,” Israeli Ministry of Health, 26 August 2021. [Online]. Available: https://data.gov.il/dataset/covid-19.

[16] D. Ayoubkhani, P. Pawelek and M. Boswort, “Prevalence of ongoing symptoms following coronavirus (COVID-19) infection in the UK,” Office of National Statistics Uk, 2 September 2021. [Online].

[17] “Preliminary dataset on confirmed cases of COVID-19,” Public Health Agency of Canada, 13 August 2021. [Online]. Available: https://www150.statcan.gc.ca/n1/pub/13-26-0002/132600022020001-eng.htm.

[18] B. Hungund, “COVID-19 Symptoms Checker,” [Online]. Available: https://www.kaggle.com/iamhungundji/covid19-symptoms-checker.

[19] B. Hungund, “COVID-19 Symptoms Checker,” [Online]. Available: https://www.kaggle.com/iamhungundji/covid19-symptoms-checker.

[20] “Python,” [Online]. Available: https://www.python.org/.

[21] Google, “Federated Learning for image classification,” 01 2021. [Online]. Available: https://github.com/tensorflow/federated/blob/main/docs/tutorials/federated_learning_for_image_classification.ipynb.

[22] CS-savvy, “Tensorflow Federated Tutorial,” [Online]. Available: https://github.com/CS-savvy/experiment_with_tf_federated/blob/master/train_tff.py.

[23] H. Taud and J. Mas, “Multilayer perceptron (MLP),” *Geomatic Approaches for Modeling Land Change Scenarios,* 2018.

[24] S. Albawi, T. A. Mohammed and S. Al-Zawi, “Understanding of a convolutional neural network,” *International Conference on Engineering and Technology (ICET),* pp. 1-6, 2017.

[25] S. Doshi, “Various Optimization Algorithms For Training Neural Network,” [Online]. Available: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6.

[26] Sanghvirajit, “A Complete Guide to Adam and RMSprop Optimizer,” [Online]. Available: https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be.

[27] “Machine Learning - Logistic Regression,” [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm.

[28] C. Maklin, “Support Vector Machine Python Example,” [Online]. Available: https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8.

[29] D. M. J. Garbade, “Understanding K-means Clustering in Machine Learning,” [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.

[30] A. Bhandari, “Everything you Should Know about Confusion Matrix for Machine Learning,” [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/.

[31] B. Hungund, “COVID-19 Symptoms Checker,” [Online]. Available: https://www.kaggle.com/iamhungundji/covid19-symptoms-checker.

[32] I. Feki, S. Ammar, Y. Kessentini and K. Muhammad, “Federated learning for COVID-19 screening from Chest X-ray images. Applied Soft Computing,” 2021.

[33] N. El-Rashidy, A. Samir , A. Tamer , A. Eslam , A. Farman , H. Jong-Wan and E.-S. Shaker , “Comprehensive survey of using machine learning in the COVID-19 pandemic,” *Diagnostics,* 2021.

[34] P. Podder, B. Subrato , . H. M. M. Rubaiyat and K. Utku , “Application of Machine Learning for the Diagnosis of COVID-19.,” *Data Science for COVID-19,* pp. 175-194, 2021.

[35] A. Hamed, S. Ahmed and N. Hamed , “Accurate Classification of COVID-19 Based on Incomplete Heterogeneous Data using a K NN Variant Algorithm,” *Arabian Journal for Science and Engineering,* 2021.

[36] A. M. U. D. Khanday, R. Syed Tanzeel, K. Qamar Rayees , R. Nusrat and M. M. U. Din, “Machine learning based approaches for detecting COVID-19 using clinical text data,” *International Journal of Information Technology,* 2020.

[37] G. Guo , Z. Liu , S. Zhao , L. Guo and T. Liu, “Eliminating indefiniteness of clinical spectrum for better screening COVID-19,” *IEEE Journal of Biomedical and Health Informatics,* 2021.

[38] S. Kang , W. Peng , Y. Zhu , S. Lu , M. Zhou , W. Lin and W. Wu , “Recent progress in understanding 2019 novel coronavirus (SARS-CoV-2) associated with human respiratory disease: detection, mechanisms and treatment.,” *International journal of antimicrobial agents.,* 2020.

[39] A. Subasi, Practical Machine Learning for Data Analysis Using Python, 2020.

[40]

[41] M. J. Taud H.