

# Analyzing the Security of e-Health Data based on a Hybrid Federated Learning Model

by

Md. Mehtabul Islam Shafin

19101088

Sabrin Akhter

18301098

Mohammad Shafkat Hasan

19101077

Md. Nasimuzzaman

19101051

Tamzeedur Rahman Prithul

18301289

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 2023

© 2023. Brac University  
All rights reserved.

# 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.

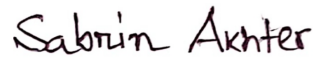
**Student's Full Name & Signature:**



---

Md. Mehtabul Islam Shafin

19101088



---

Sabrin Akhter

18301098



---

Mohammad Shafkat Hasan

19101077



---

Md. Nasimuzzaman

19101051



---

Tamzeedur Rahman Prithul

18301289

# Approval

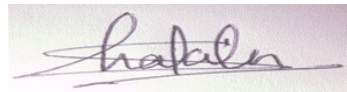
The thesis/project titled “Analyzing the Security of e-Health Data based on a Hybrid Federated Learning Model” submitted by

1. Md. Mehtabul Islam Shafin (19101088)
2. Sabrin Akhter (18301098)
3. Mohammad Shafkat Hasan (19101077)
4. Md. Nasimuzzaman (19101051)
5. Tamzeedur Rahman Prithul (18301289)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 17,2023.

## Examining Committee:

Supervisor:  
(Member)



---

Shakila Zaman

Senior Lecturer  
Computer Science and Engineering  
BRAC University

Co-Supervisor:  
(Member)



---

Dr. Muhammad Iqbal Hossain

Associate Professor  
Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)

---

Dr. Md. Golam Rabiul Alam  
Professor  
Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)

---

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

# Abstract

This research aims to provide an approach for analyzing the security of the e-health care system through the use of federated learning and the pre-processing of distinct deep learning models. The infrastructure for e-healthcare services is being gradually deployed by the health sector. This method increased the safety of patients and doctors through a protected platform. As a result, it is going to replace the current health service. Even if this technology is becoming more and more widespread, a number of data security threats need to be tackled. In this research, a CNN and MLP architecture with a classification-focused approach using a number of pre-trained feature extractors such as ResNet-50, VGG16, and Inception- v3 have been implemented. Additionally, various machine learning classification algorithms (such as Random Forest, and Logistic Regression) have been used to classify the images in order to compare how well they perform to a neural network approach. Federated learning has also been incorporated to increase healthcare data security as it does not transmit actual data but models. The objective is to develop a hybrid federated learning model to analyze the security of e-health data. The core premise is to utilize a methodology like federated learning, which enables a technique for creating machine learning models while safeguarding user privacy and can maintain e-health data security without transferring real-world data.

**Keywords:** Federated Learning; Machine Learning; e-Health Care; CNN; MLP; Random Forest; Logistic Regression;

# Dedication

We dedicated this thesis to our parents.

## **Acknowledgement**

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

Secondly, to our supervisor Shakila Zaman ma'am and co-supervisor Dr. Muhammad Iqbal Hossain sir for their kind support and advice in our work. They helped us whenever we needed help.

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

# Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Research Problem . . . . .	2
1.3 Research Objective . . . . .	3
1.4 Thesis structure . . . . .	3
<b>2 Literature Review</b>	<b>5</b>
2.1 Related Works . . . . .	5
2.2 Background study . . . . .	11
2.2.1 CNN . . . . .	12
2.2.2 MLP . . . . .	14
2.2.3 Feature Extraction . . . . .	15
2.2.4 Machine Learning classifiers . . . . .	18
<b>3 Methodolgy</b>	<b>21</b>
3.1 Research Approach . . . . .	21
3.2 Data Pre-processing . . . . .	22
3.2.1 Dataset . . . . .	22
3.2.2 Dataset Collection . . . . .	23
3.2.3 Rescale And Resize . . . . .	23
3.2.4 Encoding . . . . .	24
3.2.5 Data Augmentation . . . . .	24



3.3	Federated Learning . . . . .	24
3.4	Proposed Model . . . . .	25
3.4.1	Stochastic Gradient Descent . . . . .	26
3.4.2	SGD Algorithm . . . . .	27
3.4.3	FedAVG . . . . .	27
3.4.4	FedMAX . . . . .	28
3.4.5	FedSVRG . . . . .	28
<b>4</b>	<b>Implementation and Result</b>	<b>30</b>
4.1	Import data from Kaggle . . . . .	30
4.2	Using the CNN classifier, analyze the performance of the different pre-trained models . . . . .	30
4.2.1	ResNet50 . . . . .	30
4.2.2	VGG16 . . . . .	31
4.2.3	InceptionV3 . . . . .	32
4.2.4	Comparison of the accuracy loss plot for (i) VGG16, (ii) ResNet50, and (iii) InceptionV3 . . . . .	33
4.2.5	Analysis of the three feature extractors' performance using the random forest classifier . . . . .	35
4.3	Implementation of Federated learning Algorithm . . . . .	37
4.3.1	Creating Clients . . . . .	37
4.3.2	Aggregation through Federated Averaging . . . . .	37
4.4	Result . . . . .	38
4.4.1	Previous Results . . . . .	38
4.4.2	Comparison with related works . . . . .	39
<b>5</b>	<b>Conclusion and Future Work</b>	<b>41</b>
	<b>Bibliography</b>	<b>45</b>

# List of Figures

2.1	CNN Architecture . . . . .	13
2.2	Schematic diagram of MLP . . . . .	15
2.3	Structure of an Inception ResnetV2 layer . . . . .	16
2.4	ResNet-50 Model Architecture . . . . .	17
2.5	VGG16 Model Architecture . . . . .	17
2.6	InceptionV3 Model . . . . .	18
2.7	Random forest classifier Tree . . . . .	19
2.8	logistic regression plot . . . . .	20
3.1	Methodology Model . . . . .	21
3.2	Representation of dataset1 . . . . .	22
3.3	Representation of dataset2 . . . . .	22
3.4	Dataset Collection . . . . .	23
3.5	Dataset Collection . . . . .	23
3.6	Federated Learning (FL) model. . . . .	25
3.7	Working Mechanism of FL . . . . .	26
3.8	Stochastic Gradient Descent has gone down a certain route. . . . .	27
4.1	Import data from Kaggle . . . . .	30
4.2	ResNet50 Validation Accuracy Plot . . . . .	31
4.3	VGG16 Validation Accuracy Plot . . . . .	31
4.4	InceptionV3 Validation Accuracy Plot . . . . .	32
4.5	Accuracy loss plot for (i) VGG16 . . . . .	33
4.6	Accuracy loss plot for (ii) ResNet50 . . . . .	34
4.7	Accuracy loss plot for (iii) InceptionV3 . . . . .	34
4.8	Simple MLP . . . . .	35
4.9	Simple CNN . . . . .	36
4.10	CNN Accuracy Graph . . . . .	36
4.11	Creating Clients . . . . .	37
4.12	Aggregation through Federated Averaging . . . . .	38

# List of Tables

4.1	Validation accuracy for different epoch counts. . . . .	33
4.2	Validation accuracy for different Random Forest. . . . .	35
4.3	Summary of Results . . . . .	38
4.4	Results of 10 epochs training with the learning rate of 0.01. . . . .	39
4.5	Results of 50 epochs training with the learning rate of 0.01. . . . .	39
4.6	Comparison with related works . . . . .	40

# Nomenclature

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

*CNN* Convolutional Neural Network

*FedAVG* Federated Averaging

*FedMax* Federated Maximum

*FedProx* Federated Proximal

*FedSVRG* Federated Stochastic Variance Reduced Gradient

*FL* Federated Learning

*LR* Logistic Regression

*MLP* Multi Layer Perceptron

*SGD* Stochastic Gradient Descent

*SVM* Support Vector Machine

# Chapter 1

## Introduction

### 1.1 Overview

In the last few decades, the use of technology in health and healthcare has advanced exponentially. The medical community has utilized technology in a variety of ways, including imaging techniques for diagnosis, electronic health records, robotics in surgical procedures, telehealth to reduce geographical and temporal barriers between patients, and wearables for monitoring individual health [40]. For example, in the field of genomics, where data regarding genetic makeup, biomarkers, and bioinformatics are used to create more effective therapeutic solutions, the use of open data sources is equally crucial. As the healthcare industry continues to immerse itself in technology and its many subfields, such as artificial intelligence (AI) and machine learning (ML), the importance of how it utilizes and secures this data grows [42]. The replacement of handwritten data with electronic health records in order to preserve the constant flow of patient-related data, including personal information, diagnosis, treatment, and follow-ups, is one area where data security becomes a major problem. The majority of these repositories are hosted on open-source, easily accessible, and downloadable platforms. With this ease of access to patients' information, it becomes even more crucial to protect their data to prevent improper handling of private and sensitive data.

Federated learning is a strategy for distributed machine learning in which multiple participants, such as devices or enterprises, collaborate to train a model while retaining local access to their data. Federated learning is a paradigm for learning that aims to address the problem of data governance and privacy by training algorithms collaboratively without exchanging the data itself. It enables model training on huge and varied datasets and enables privacy-preserving machine learning [41]. It was originally developed for various domains, such as mobile and edge device use cases, but has recently gained popularity for the healthcare sector. FL enables collaboratively gaining insights in the form of a consensus model without moving patient data outside the firewalls of the institutions where they reside. Recent research has demonstrated that models trained by FL can achieve performance levels comparable to those of models trained on centrally hosted data sets and superior to those trained on single-institution data sets [9].

In 2017, Google completely detached the capability to perform machine learning

(ML) from the requirement to store data in the cloud by introducing federated learning (FL), a methodology that simply allows devices to collaboratively train machine learning (ML) models while maintaining the raw training data on every browser[4]. Federated learning allows users to simultaneously train models across all of their data without storing their data in many locations. This indicates that compared to training it on a single computer, it can be trained more swiftly with less computing or storage resources required from each location. The fundamental aspect of “data privacy” is the cornerstone of the upcoming development of AI technology. When data security is of extreme significance and there is no room for ambiguity, federated learning can be used to preserve data transfer that safeguards our personal information by developing intelligent systems in secret.

## 1.2 Research Problem

The healthcare industry is becoming increasingly reliant on technology. The security of data in e-health is a top priority as the data of patients and doctors contain sensitive information. With the COVID-19 pandemic, more people are using e-healthcare services, making them more vulnerable to online attacks. In 2015, researchers found that over 68,000 medical systems were exposed online, many of which were running outdated and insecure operating systems such as Windows XP[34]. These systems were easy to hack using simple techniques, and researchers found that attackers were able to leave malware payloads behind on specialized servers posing as medical devices. The need for strong and secure login techniques is essential in the modern healthcare industry.

Researchers have found that it was simple to find and hack medical equipment connected to the internet using obsolete and vulnerable systems using the search engine Shodan and basic tactics. To solve the problem, two security experts created honeypots, specialized servers disguised as medical equipment with secure login procedures. The honeypots showed that attackers were able to access the machines over 55,000 times and leave malware payloads behind. The healthcare sector has long been a target of hackers. The size and frequency of healthcare data breaches have grown over the last five years, impacting almost 80 million people. Breach of healthcare data routinely exposes highly sensitive information. Patient’s social security numbers, names, and addresses are all vital information. Other vital elements include medical identification, insurance information, and medical histories[1].

According to the Office of Civil Rights of the US Department of Health and Human Services, the top 10 healthcare data breaches (ranked by the number of individuals affected) are as follows: The first is Newkirk, which had 3.47 million affected things as of August 2016, followed by Banner Health, which had 3.62 million damaged items at the same time. Medical Informatics Engineering had 3,9 million Affected products as of June 2017, followed by Advocate Health Care with 4,03 million (August 2013). With 4.5 million affected items, Community Health Systems and the University of California, Los Angeles Health are ranked sixth and fifth, respectively. Prior to September 2011, 4,9 million people were covered by TRICARE. Premera Blue Cross, Excellus BlueCross BlueShield, and There are 78.8 million goods af-

fectured by Anthem Blue Cross, respectively[7].

Medical data sets are hard to come by, though. FL solves this problem by promoting cooperative learning without centralized data storage. The goal of federated efforts (FL) is straightforward: by permitting ML from non-co-located data, it will be possible to address issues with privacy and data governance. Each data controller in an FL context establishes its own governance procedures and privacy guidelines, as well as managing data access and having the authority to revoke it. In electronic health records (EHR) and medical imaging, FL can be used to represent and locate patients who are clinically similar. Large-scale alliances like the Trustworthy Federated Data Analytics project are setting the bar for future norms of innovative, safe, and safe collaboration in healthcare applications.

### 1.3 Research Objective

This study aims to analyze the security of the e-healthcare system using hybrid federated learning models and the pre-processing of diverse deep learning models. The objective is to explore the security-enhancing system in e-health. In this research, CNN and MLP models have been implemented. To train these models, a variety of feature extraction strategies, including resNet50, VGG16, and Inception-v3 have been used. Therefore, Machine learning classifiers are employed to improve accuracy and identify the optimal model. Eventually, federated learning techniques are utilized to ensure the security of healthcare information.

The system will also use hybrid federated learning models , allowing many parties to work together and develop a common model while maintaining local control over their own data. The system is protected from data breaches and illegal access. The research will also assess the effectiveness of the system and offer recommendations for improvement. The study will ultimately provide a safe system that can efficiently gather and categorize sensory input from medical devices while also guaranteeing the confidentiality and privacy of the collected data.

### 1.4 Thesis structure

The first chapter provides an overview of the e-health care system using hybrid federated learning model and its applications in the healthcare industry, particularly for data security. It has covered the security and privacy concerns, as well as its growing popularity in the healthcare sector. In addition, this chapter briefly discusses the challenges the healthcare industry faces as a result of technological advancement and introduces the federated learning approach as a solution.

The second section of chapter 2 consists of related works and a comprehensive analysis of the context. This section describes a study that uses CNN and MLP models to classify input images into six categories after extracting image features using pre-trained deep learning models such as VGG, ResNet, and Inception. It also specifies

that the healthcare dataset is classified using machine learning techniques and that the security of patient health information is enhanced using FedAVG, FedMAX, FedSVRG, and FedPROX.

Chapter 3 includes the methodology, data preprocessing steps, and proposed model. Using this research methodology, data is collected, pre-processed, pre-trained models are selected for feature extraction, the dataset is divided into train and test groups, a classification-focused CNN and MLP architecture is implemented, and the CNN classifier is trained for image classification. Federated Learning is also used to evaluate the model's accuracy and prevent data transfer to increase the security of healthcare data.

In the fourth chapter, the implementation, and results of the proposed model are discussed. This chapter focuses on analyzing the performance of various pre-trained models by training a CNN classifier on a dataset obtained from Kaggle and pre-processing the images. The results are presented in tables and figures that illustrate the validation accuracy of the models for varying numbers of training epochs. A Federated learning algorithm also improves the security of patient health information by minimizing data transfer and model communication.

The fifth chapter concludes with a summary and outlook for the future. The study suggested to improve patients' data accessibility and using federated learning on a hybrid database to improve healthcare data security. Improving model performance relative to centralized models and addressing device heterogeneity in federated learning is, therefore, some of the limitations and future research areas.



# Chapter 2

## Literature Review

### 2.1 Related Works

E-Health is a popular term for healthcare centered on electronic technologies and interaction. E-Health comes up with comprehensive information about widespread diseases and ailments, including how to treat them globally utilizing electronic communication. e-health refers to the delivery of medical services via contemporary electronic information and communication technologies when patients and medical professionals are not in close proximity to one another and their communication is mediated by technology. These services include, among others, telepathology, vital sign monitoring, electronic prescription, teleconsultation, and physical and psychological diagnosis and treatment.

Studies have shown that distributed learning models (FL) are at least as accurate as centralized learning models. Data analysis from the Internet of Health Things devices with little resources can be done using FL. The mental condition of patients can also be predicted using physiological and mobility data gathered by medical IoT devices. The application of well-known distributed designs to the FL problem is a major goal of this work. Some theories rely on using regional datasets to assess the precision of regional and global models. Others look into the effectiveness of homomorphic encryption to safeguard learning models on FL used in healthcare applications. Studies that aim to demonstrate the secrecy of data and ML models rely on integrating homomorphic encryption with additional methods. The review of the literature indicates that in order to protect federated learning architectures against potential assaults, FL needs to be extensively developed. In the literature, there was more attention than usual on outcome prediction in particular[12].

A model that has gained favor recently is federated learning. It enables maintaining the data in local institutions while training a common global model with a central server. The authors describe the federated learning setup, discuss common issues and solutions, and consider potential uses in the medical field. Multitask learning (MTL) is a logical method for handling data from several distributions. In comparison to AFL, q-Fair Federated Learning gives underperforming devices more weight. The generalizability of results is constrained by systemic and random biases that are present across hospitals and generally. Since healthcare data is extremely sensitive and its use is strictly controlled, data exchange in this industry is uncommon.

Without actually sharing the data, federated learning makes an effort to address the problems with data governance and privacy. A workable instrument for data-driven medicine is the federated modeled approach (FL), which has the potential to provide precision medicine on a wide scale. If FL is successfully implemented, judgments could be made that respect privacy concerns and take into account rare conditions. Despite the fact that ML and DL are rapidly being utilized as the norm for knowledge discovery across a variety of industries, the successful implementation of data-driven applications calls for a significant and diverse dataset[2].

The Mobile Healthcare Monitoring Network (MHMN), which allows users to track their health and receive pre-diagnoses online, is also discussed. MHMN can monitor a patient’s health in real-time without disrupting their daily activities. On the other hand, health professionals are worried about the security of private information. Sensors capture personal data from patients every second and send it to the cloud server as MHMN multi-dimensional vectors. When real-time data is incorrect, the cloud server saves personal information and sends monitoring data to the hospital. Researchers showed how to swiftly obtain MHMN’s desired characteristics while ensuring data privacy. They compared their results to recent advancements in pre-diagnosis online privacy-preserving technology and PHR search outsourcing. Recent studies have not shown how effectively textual multi-keyword ranked searches and digital vector range queries can coexist. CS may offer HU textual multi-keyword ranked search services as well as range searches that protect privacy. In order to diagnose illnesses without jeopardizing PU’s privacy, HU might train classifiers using a sizable medical dataset that is maintained in CS[20].

A secure network for exchanging medical data can be developed with the help of blockchain technology. An investigation revealed that there have been more medical records leaked annually. Privacy standards require that data be handled, distributed, and stored in a safe and private way. It has long been challenging to share medical data while yet adhering to security and privacy regulations. Many alternative strategies may be required for a practical approach to medical data exchange. Blockchain provides advantages over previous computer paradigms as a new computing paradigm. It is critical to select the appropriate blockchain type (permission or permissionless)[13].

The Internet of Things (IoT) is a network of billions of physical items that connect to the internet and perform tasks mostly independently of people. IoT networks become an open source of personally identifiable information when routine tasks are cleverly automated, making it possible for malicious attackers to steal, change, and utilize this data for their own bad ends. Machine learning (ML) assisted techniques have recently attracted a lot of attention in the area of IoT security. In much recent research, it is assumed that enormous training data is widely accessible and transferable to the main server since data is continuously produced by IoT devices at the edge. This means that because it depends on the legacy set of all data that is kept on a single server, classic ML is the least preferable option for domains with privacy concerns over user data. We propose a method to address this issue by proactively detecting infiltration in IoT networks using decentralized on-device data in a federated learning (FL)-based anomaly detection strategy. Our method em-

employs federated training cycles on gated recurrent unit models and only distributes the learned weights to the FL's main server, allowing the data to remain on local IoT devices. Additionally, to improve the accuracy of the overall ML model, this part of the approach mixes updates from many sources. Our testing results demonstrate that our approach provides the best accuracy rate for attack detection and protects user data privacy better than traditional/centralized machine learning (non-FL) versions[6].

Due to the growing concerns about sharing personal data and the growing processing capability of wireless end-user equipment (UE), a new machine learning (ML) paradigm known as federated learning (FL) has emerged. In particular, FL enables the decoupling of data provision to UEs and ML model aggregation at a central unit. By creating the model locally, FL is able to stop direct data leakage from the UEs, thus preserving some level of security and privacy. Additionally, some recently discovered attacks against the FL architecture can still obtain a person's private information even if raw data are not disclosed via UEs. In this paper, the privacy and security issues in FL are discussed, along with the challenges of securing privacy and security when developing FL systems. In-depth simulation data is also represented in order to illustrate the difficulties and solutions that have been raised[32].

It is found that PCML, which has been proposed in various studies, is a new skyline-based collaborative model learning technique that protects user privacy. Each healthcare center's sensitive medical data is adequately protected since PCML can be used to learn a global diagnosis model while maintaining the security of its local diagnosis models. Online medical diagnostics is versatile and practical since it eliminates geographical restrictions and reduces the amount of time spent waiting to see a doctor. Data mining methods have recently been created for the e-healthcare system. There are still difficulties with the collaborative learning model used by many healthcare facilities. Numerous privacy-preserving techniques, including homomorphic encryption and anonymity algorithms, have been created to achieve data security. The majority of them require significant computing resources due to their huge, sophisticated mathematical operations. Many healthcare organizations can utilize PCML to learn a more precise global diagnosis model using their local diagnosis models[14].

A promising method for creating precise and reliable statistical models from the vast amounts of medical data that are gathered by contemporary healthcare systems is data-driven machine learning (ML). Due to data silos and access restrictions brought on by privacy concerns, existing medical data is not completely utilized by machine learning (ML). But without adequate data, ML won't be able to realize its full potential and finally go from the realm of academic study to the realm of clinical application. This essay examines the main causes of this problem, evaluates how federated learning (FL) can offer a remedy for the future of digital health, and emphasizes the difficulties and issues that need to be taken into account[23].

Numerous studies highlight the potential of deep learning in discovering complicated patterns that can result in biomarkers for diagnosis and prognosis. It can be difficult and infrequently discovered in particular institutions to locate sufficiently vast

and diverse datasets, which are needed for training. Privacy and ownership issues arise in multi-institutional collaborations based on centralized patient data sharing. By spreading the model-training to the data-owners and aggregating their outputs, federated learning is a revolutionary paradigm for data-private multi-institutional cooperation that makes use of all accessible data without sharing data between institutions. That assess generalizability using data from non-federated institutions. For further examine how data distribution among cooperating institutions affects model quality and learning patterns, indicating that greater access to data through multi-institutional collaborations that are data private can improve model quality more than the errors that the collaborative approach introduces. Finally, it compares federated learning to various collaborative learning strategies to show its superiority and talk about real-world implementation issues. The use of federated learning in clinical settings is anticipated to result in models being trained on datasets of unprecedented size, which will accelerate the development of precision/personalized medicine[15].

Since data is born at the edge and continuously produced by IoT devices, huge data containing personal Identifiable Information is available and transferable to main server. This is where Machine Learning steps in. Since, it works on the legacy set of entire data it is not preferred for domains with privacy concerns on user data. In this paper the authors propose that Federated Learning based anomaly direction approach to proactively recognize trespassers in IoT networks using decentralized on device data. This is as federated training rounds on Gated recurrent Units models and keeps the data intact on local IoT devices by sharing only the learned weights with the central server of the FL. At the same time, approach's ensembles part aggregates the update from multiple sources to optimize the global Machine Learning models accuracy[33].

Large-scale machine learning challenges require computationally effective and privacy-aware solutions in the era of "big data," especially in the healthcare industry where vast amounts of data are housed across several sites and are the property of various parties. Previous studies have concentrated on centralized algorithms, which presuppose the existence of a central data repository (database) that can store and process data from all users. A single point of failure risk introduced by such an architecture can jeopardize the security and privacy of the data and make it impracticable when data are not centralized. In addition, it does not scale well to very large datasets. A decentralized computationally scalable methodology is desperately needed due to the vast amounts of data that are dispersed between hospitals and individuals[8].

Deep learning has led to numerous successful smart healthcare applications that use data to improve clinical institutions' treatment. Data-driven deep learning models work well. More data taught makes the deep learning model more robust and generalizable. To develop a viable deep learning model, The authors explain that medical data must be centralized. Privacy, ownership, and tight regulation issues arise. Federated learning handles the problems with a global deep learning model and a central aggregator server. The local party secures and anonymizes patient data. First, the author review federated learning in healthcare research. Second, They assess new federated learning difficulties from a data-centric perspective, in-

cluding data partitioning features, data distributions, data protection measures, and benchmark datasets. Furthermore, authors discuss healthcare application issues and future research[36].

In the article, the problems with and solutions for communication latency and cost in federated learning are examined. In order to reduce the number of iterations between participants and the server while maintaining accuracy, the authors suggest a Federated Stochastic Variance Reduced Gradient technique. When compared to conventional stochastic gradient descent-based Federated Learning, the suggested method can significantly lower the communication cost, according to simulation results on issues involving linear and logistic regression[30].

Brain images of numbers that were previously inconceivable are currently present in databanks all across the world. These enormous amounts of data have the potential to help us understand the genetic roots of brain illnesses when combined with advances in data science. The entire potential of big data for the study of brain illnesses is, however, constrained by privacy and regulatory issues that prevent the direct sharing of various datasets that are kept at various institutions. Here, it is suggested using a federated learning architecture for anonymously accessing and meta-analyzing any biomedical data. By examining brain structural correlations across illnesses and clinical cohorts, which demonstrate the theory. The framework is initially put to the test on fictitious data before being used in multi-centric, multi-database studies including ADNI, PPMI, MIRIAD, and UK Biobank. This demonstrates the framework’s potential for future use in distributed analysis of multi-centric cohorts[11].

The authors use iterative model averaging to implement federated deep network learning and evaluate five model architectures and four datasets[5]. These studies show the technique is resistant to the imbalanced and non-IID data distributions that define this scenario. They reduce communication rounds by 10-100x compared to synchronized stochastic gradient descent, which is the main restriction.

Federated Learning (FL) over the Internet of Medical Things (IoMT) devices is now an area of research that is getting a lot of attention. In this paper, the authors propose FEDMSQE – Federated Learning with Minimum Square Quantization Error, which gets the smallest quantization error for each client in the FL setting[39]. This is done to answer the questions above. They demonstrate that the proposed algorithm is more accurate and has less quantization error than other quantization methods by running numerical tests in both single-node and FL scenarios.

Federated learning is a new machine learning technique that enables collaboration among all participants while maintaining data privacy in order to train a global optimal machine learning model[35]. Federated learning is viewed as a solution to the data privacy issue while still allowing the combining of similar or related data from various sites all over the world by allowing the sharing of machine learning models rather than raw data. This article offers a thorough explanation of the idea of federated learning and how it can be used with private healthcare datasets.

The research shows that every local hospital in a collaborative training system can add concealed backdoor functionality to the global model. For clean inputs, the backdoored joint global model behaves properly, but for predetermined triggers, it produces an adversary-expected outcome. FL’s decentralized nature makes detect-

ing backdoor assaults difficult. The authors offer a real-time FL backdoor detection technique using a coalitional game and Shapley value[38]. Extensive trials on two machine learning tasks show that our methods are accurate and robust to multi-attackers.

This research proposes a novel master-key management technique to manage keys and improve healthcare information security, and the multi-key server approach balances key server load for fast access to electronic health records. The authors build a novel large-scale e-healthcare group key management system architecture to give secure and reliable assistance to healthcare firms and service providers[21]. The researchers tested in a group of healthcare users joining and leaving dynamically and validated by applying it to virtual servers, showing that rekeying overhead, risks, and computing costs of the healthcare network system are minimized compared to existing approaches.

In e-healthcare systems where sensitive patient data is outsourced to data centers, the article offers a potential strategy for securely and confidentially querying clinical paths. A variety of sub-protocols are used in the scheme to secure system privacy, and a greedy algorithm and [27] Min-Heap technology are used to increase efficiency. The experimental findings demonstrate how effective and useful the proposed approach is for clinical pathway inquiries.

This study examines how Jordanian e-health systems improve medical staff performance, patient care, and physician-patient interactions. The authors analyze a single integrated technology acceptance model. This study employed logical research. 212 medical staff from 19 Jordanian hospitals provided data. A partly square/structural equation modeling method was used to examine data and test study hypotheses[28]. The researchers examine HIPAA's medical record privacy and security provisions[3]. HIPAA regulates PHI and its security. A health data entity must observe privacy regulations to protect patients' confidential medical information. The HIPAA security regulation requires physical, technical, and administrative precautions to preserve PHI privacy. This article reviews HIPAA's history, rules, ramifications, and imaging professionals' roles.

Researchers use a privacy-preserving method to classify multi-site fMRI. They offer federated learning using a decentralized iterative optimization method and a randomization mechanism to change shared local model weights[19]. This federated learning framework proposes two domain adaptation approaches to account for site-specific fMRI distributions. They study federated model optimization and compare it to other training methods. Their findings suggest that multi-site data without data sharing can improve neuroimage processing and identify disease-related biomarkers.

Authors suggest big data, federated learning systems, particularly biological ones, are reviewed in this survey[43]. Specifically, they describe the broad answers to federated learning's statistical, system, and privacy difficulties and discuss healthcare implications.

The researchers explore query inconsistencies, knowledge base gaps, and user domain information sets that impede progress in this subject. From diagnostic heart failure to 1-D cardiovascular beatings and automated detection employing multi-dimensional clinical data, machine learning and artificial intelligence have rapidly advanced medical applications. Thus, clever decision-support frameworks help doctors make better treatment judgments. An interesting option is to harness expand-

ing healthcare digitization that produces huge amounts of clinical data in e-HCR and mixes it with advanced ML algorithms to improve clinical decision-making and expand the drug evidence base. The authors examined innovative methods and real-life medical technologies, focusing on e-HCR patient portrayals[24].

The machine learning technique for heart disease diagnosis proposed in this paper is effective and precise. In the system, classification techniques include Support Vector Machine (SVM), Logistic Regression[18], Artificial Neural Network (ANN), K-Nearest Neighbor (K-NN), Naive Bays, and Decision Tree. Feature selection algorithms including Relief, Minimal Redundancy Maximal Relevance (MRMR), Least Absolute Shrinkage Selection Operator (LASSO), and Local Learning increase classification accuracy and reduce execution time. The suggested feature selection algorithm (FCMIM) with SVM classifier performed well on the Cleveland heart disease dataset.

In order to safeguard patients' privacy, the article [16] suggests a quick and secure image encryption solution. It makes use of a block symmetric encryption algorithm based on confusion and diffusion operations as well as a new random number generation approach based on two chaotic maps. The proposed encryption system has been put to the test and has been found to be more efficient and secure than the methods currently in use. It also protects the privacy of keyframes' medical information while using less energy, communication bandwidth, and specialist analysis time during the WCE procedure.

The use of machine learning and deep learning techniques in healthcare applications is discussed in this paper [22], along with the security and privacy issues that come with it. As well as emphasizing the necessity to address privacy and security concerns to allow a secure and reliable implementation of these models in clinical settings, it draws attention to how these techniques have the potential to alter the delivery of healthcare services. The report also identifies open research issues that need further study and discusses potential solutions to deliver secure and privacy-preserving ML. The most recent platforms, protocols, and network topologies that are being used in IoT-based healthcare systems [29] are all covered in this paper's overview of the state-of-the-art technologies and techniques employed in these systems. It also offers a thorough examination of the security and privacy issues that occur in these systems and suggests a security architectural model based on blockchain address them. The study also addresses IoT market prospects in healthcare and identifies areas for further investigation.

## 2.2 Background study

For feature extraction, a number of pre-trained models are available, including VGG, ResNet, Inception, and DenseNet. Although VGG16, ResNet50, and InceptionV3 have been studied in this research. While extracting features from input photos, these well-liked pre-trained deep-learning models are used. The CNN and MLP models were then put up to categorize the images into six categories: AbdomenCT, HeadCT, BreastMRI, ChestCT, Hand, and CXR. To further classify a healthcare dataset and apply FedAVG, FedMAX, FedSVRG, and FedPROX to enhance the security of patient health information, several machine learning algorithms, including stochastic gradient descent (SGD), random forest, logistic regression, and SVM, have also been used.

### 2.2.1 CNN

The convolution neural network (CNN) model was frequently used in the process of medical diagnosis. Since CNN is a great feature extractor, using it to identify medical images can minimize costly and time-consuming feature extraction. Three layers make up CNN: convolutional, pooling, and fully linked layers. The pooling layer is the next significant layer in the CNN structure. Pooling layers come in two varieties: max pooling and average pooling. This layer is primarily used to decrease the number of variables because fewer parameters make the learning process of the model easier. The fully connected layer of CNN operates as its final layer and is mostly utilized to categorize features based on the findings of the layers that came before it. Generally, the Softmax function is applied in this layer as an activation function. It employs a grid layout to examine the data in accordance with the categorization of neural networks. Most computations are handled by CNN's convolution layer, which is its base layer.

Additionally, the most often used techniques for classifying photos are SVM and ANN classifiers. These methods produce results with high levels of classification performance as well as accuracy, responsiveness, and sensitivity. A back-propagation technique with arbitrary weights  $W$  is used to train CNNs by minimizing the cost function based on the multilayer perceptron.

#### For 2D image $H$ and 2D Filter (Kernel) $F$

1. Convolution operation:

$$G = H * F \quad (2.1)$$

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v] \quad (2.2)$$

2. Correlation Operation:

$$G = H \oslash F \quad (2.3)$$

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i + u, j + v] \quad (2.4)$$

According to the convolution formula 2.1, the input image  $H$  and filter  $F$  are convolved to produce the output image  $G$ . The formula adds the products of each corresponding pair of pixels from the input picture  $H$  and the filter  $F$  for each pixel  $I j$ ) in the output image  $G$ . The area of the filter  $F$  and the equivalent area of the



input picture  $H$  are combined in the sum 2.2. The positions of the matching pixels being multiplied in the input image  $H$  are shown by the indices  $u$  and  $v$ .

Based on the correlation formula 2.3, the output picture  $G$  is created by comparing the filter  $F$  with the input image  $H$ . The formula adds the products of each corresponding pair of pixels from the input picture  $H$  and the filter  $F$  for each pixel  $(i,j)$  in the output image  $G$ . The area of the filter  $F$  and the equivalent area of the input picture  $H$  are combined in the sum . The positions of the matching pixels being multiplied in the input image  $H$  are shown by the indices  $u$  and  $v$ . The indices  $u$  and  $v$  in this formula 2.4, however, are inverted, shifting the filter  $F$  by  $(u, v)$  in the opposite direction of the input picture  $H$ .

It is part of a category of ANNs that can recognize patterns and categorize images in keeping with all those patterns. An input layer, numerous hidden layers that may include convolutional layers and non-convolutional layers, and an output layer form the fundamental components of CNN's design. The images are analyzed and patterns are predominantly found using the hidden layers. The shared weight theory supports CNN's procedures. Every layer's neurons have a weight and a threshold value, and they are linked to other neurons with the same weight. A neuron is stimulated when its output exceeds the threshold, and its output is then transmitted to the network's next layer.

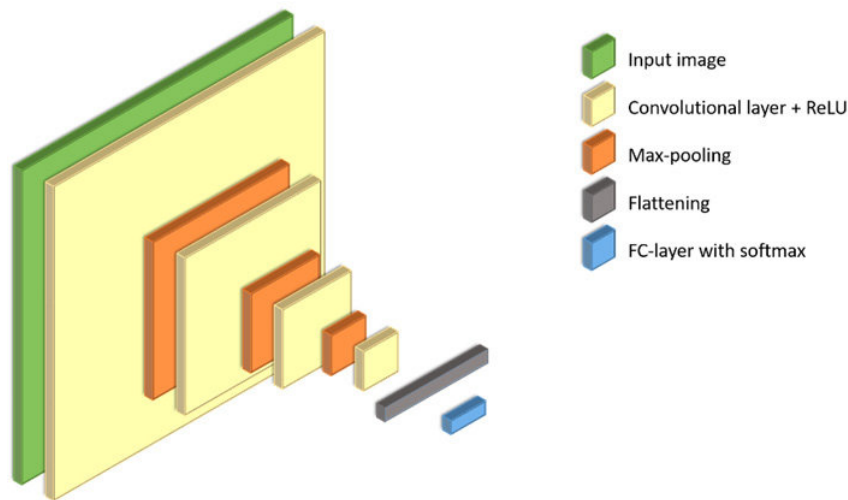


Figure 2.1: CNN Architecture

The most important layer of CNN, where the majority of the calculation takes place, is the convolutional layer. During the transformation of the input using some filters or kernels, this layer transmits the modified input to the subsequent layer. The convolution operation is the term for this input transformation technique. In order to recognize the patterns in the images 2.1 or data, we need to provide the amount and size of the kernels that will be employed during convolution. These kernels filter the input by iteratively traversing through the input data with the selected stride. A feature map is generated following the completion of the filtering procedure. ReLU functions are widely used as the activation function in this layer. The feature map is reactivated by the ReLU activation function, which zeroes out all negative values. The neuron's input must be higher than the threshold value for it to activate. As

a result of this, a neuron produces zero as its output when its input is less than zero. The function demonstrates a linear relationship with the dependent variable as illustrated in the following figure when the input exceeds the threshold value.

## 2.2.2 MLP

A feed-forward multilayer perceptron, the MLP consists of at least one input layer, one or more hidden layers, and an output layer, each of which executes a unique set of operations. Numerous neurons have an activation function for each hidden and output layer. Complex non-linear correlations between input and output factors, such as the activation function, hidden layers, and the number of neurons in each layer, can be modeled using artificial neural architecture.

### Forward Propagation in MLP

$$Z_1^{(h)} = a_0^{(in)} w_{0,1}^{(h)} + a_1^{(in)} w_{1,1}^{(h)} + \dots + a_m^{(in)} w_{m,1}^{(h)} \quad (2.5)$$

$$a_1^{(h)} = \phi(Z_1^{(h)}) \quad (2.6)$$

The equation 2.5 describes forward propagation in a multilayer perceptron (MLP). The weighted sum of the inputs, denoted by  $h$ , and the corresponding weights, denoted by  $w$ , are computed using this equation. The outcome of activating the  $z$  value is the activation unit 2.6. As the activation function 2.7, the sigmoid (logistic) function is frequently used. Using the inputs and weights as inputs, this equation determines the likelihood of the output.

$$\phi(Z) = \frac{1}{1 + e^{-z}} \quad (2.7)$$

### Feature extraction and identification using MLP

The most effective internal representation of the input signal for the classification task is created by MLP. From this perspective, the MLP executes some sort of feature extraction that is provided by the hidden units' activity levels. The majority of MLP feature extraction architectures have fewer units on the hidden layer than on the input layer. As a result, the hidden layer achieves some kind of dimensionality reduction by acting as a narrow-band channel. Again, assuming the MLP's learning process is successful, one can anticipate that this reduction extracts the signal's most prominent features.[1]

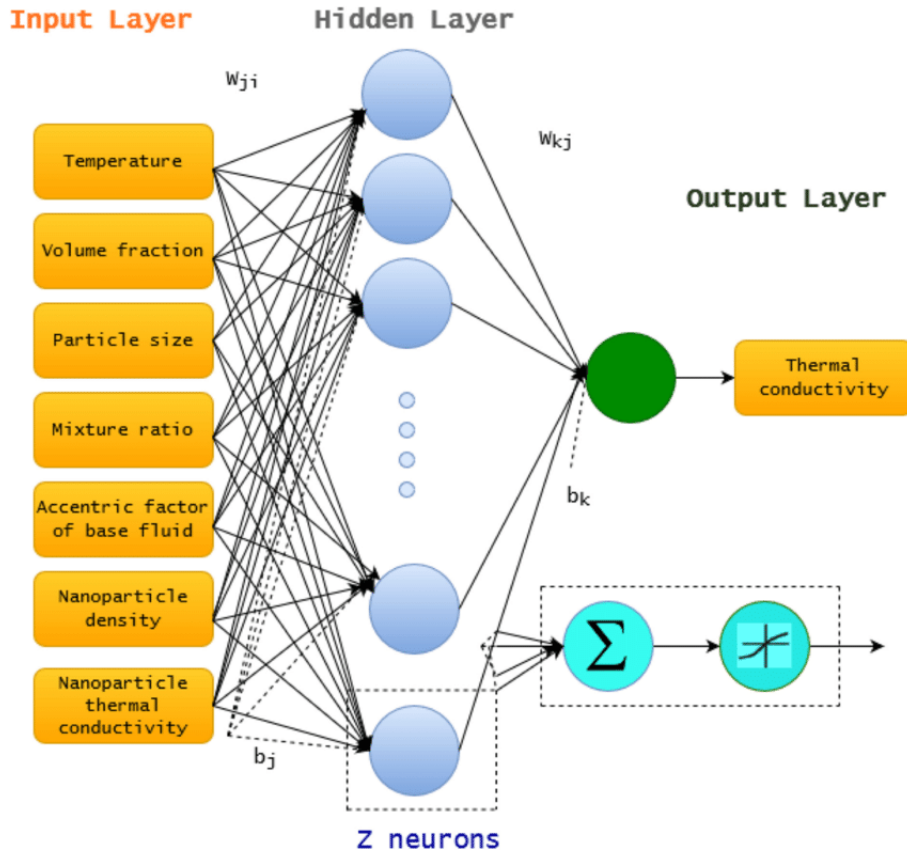


Figure 2.2: Schematic diagram of MLP

### 2.2.3 Feature Extraction

#### Inception-ResNet-v2

The ImageNet database, a collection of more than a million photos, was used to create the convolutional neural network known as Inception-ResNet-v2. A keyboard, mouse, pencil, and several more creatures are among the 1000 different object categories that the 164-layer deep network can classify images into. As a result, the network has gathered thorough feature representations for a variety of photos. Images with a resolution of 299 by 299 are accepted by the network.

Pre-trained Deep Neural Networks in MATLAB include additional pre-trained networks. The Inception-ResNet-v2 network 2.3 can be used to classify and categorize fresh pictures. Replace GoogLeNet with Inception-ResNet-v2 and then proceed as directed in Classify Image Using GoogLeNet. When utilizing the Train Deep Learning Network to Classify New Images technique to retrain the network for a new classification job, Inception-ResNet-v2 should be loaded in place of GoogLeNet.

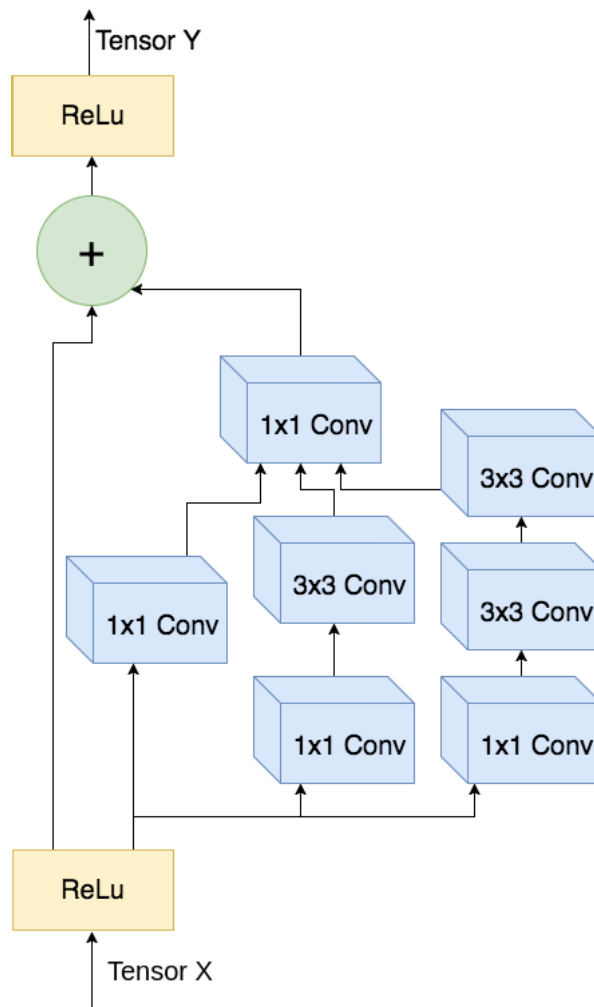


Figure 2.3: Structure of an Inception ResnetV2 layer

## ResNet-50

Convolutional neural network ResNet-50 has 50 layers overall. Users can load a pre-trained version of the network from the ImageNet database that has been trained on more than a million images. The trained network can classify images into 1000 different object categories, which include a variety of animals, a mouse, a keyboard, and a pencil. As a result, for many different photos, the network has gathered rich feature representations. The network will accept photos with a resolution of 224 by 224. See Pre-trained Deep Neural Networks for other pre-trained networks in MATLAB.

The ResNet-50 model 2.4 can be used to classify fresh photos. Replace GoogLeNet with ResNet-50 and then proceed as directed in Classify Image Using GoogLeNet.

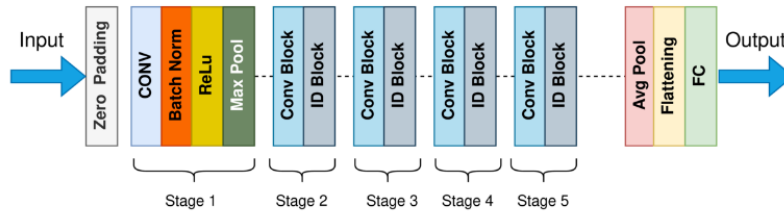


Figure 2.4: ResNet-50 Model Architecture

## VGG16

The VGG16 architecture of a convolution neural network (CNN) 2.5 was used to win the ILSVR (Imagenet) competition. One of the greatest visual field model architectures available today, according to several experts. The most notable feature of VGG16 is that, rather than emphasizing the usage of a large number of hyper-parameters, they focused on having convolution layers of 3x3 filter with a stride 1 and consistently employing the same padding and max pool layer of 2x2 filter with a stride 2.

Convolution and max pool layers are placed in this manner throughout the system. It has two fully connected layers (FC) in the end, and a softmax is used for output. The 16 in VGG16 indicates that there are 16 weighted layers.

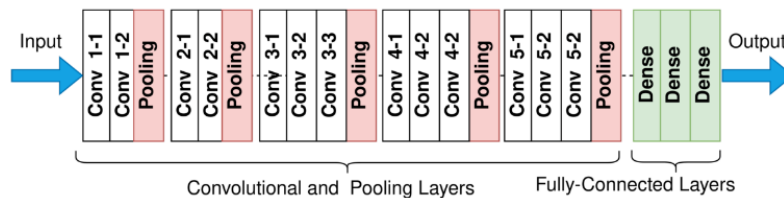


Figure 2.5: VGG16 Model Architecture

## Inception-v3

Neural network using convolutions To help with object classification and picture analysis, Inception v3 2.6 was created as a Googlenet plugin. This iteration of the Google Inception Convolutional Neural Network is the third since it was first introduced for the ImageNet Recognition challenge. Deeper networks can be supported by Inceptionv3 while keeping parameter number expansion to a minimum. Compared to 60 million for AlexNet, it has “under 25 million parameters.”

Similar to how ImageNet may be thought of as a database of classed visual objects, Inception aids in the classification of items in the field of computer vision. The Inceptionv3 architecture has been used by many applications, and it is regularly combined with ImageNet’s “pre-trained” data. It has applications in the life sciences, such as leukemia research.

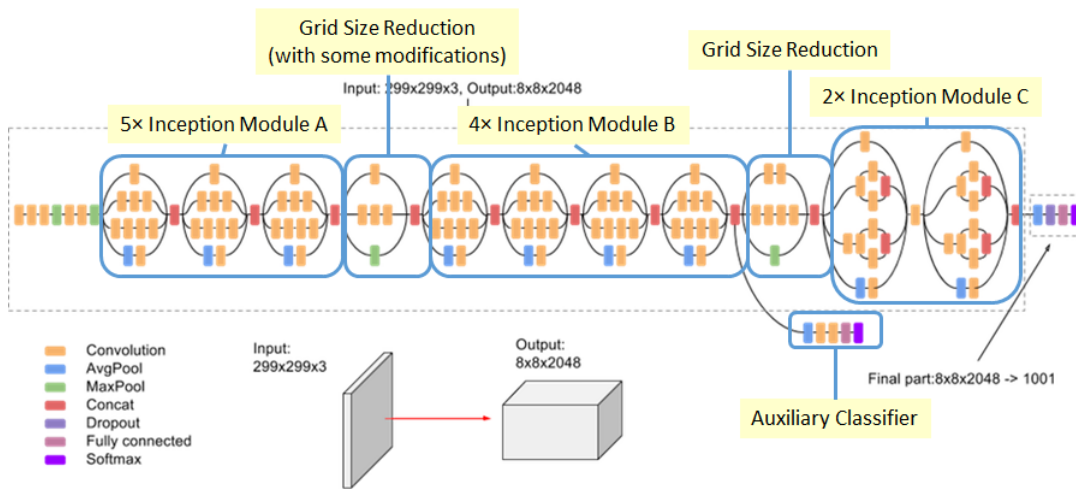


Figure 2.6: InceptionV3 Model

## 2.2.4 Machine Learning classifiers

### Random forest

Random Forest consists primarily of a large number of independent decision trees. The random forest method 2.7 is a better version than an ordinary decision tree because there is a high chance of getting a high variance result. As an ordinary decision tree algorithm is quite sensitive to training data it may not succeed in generalizing. On the other hand, training data is less sensitive in the Random Forest Algorithm because this algorithm provides uncorrelated results so the chance of getting a more accurate response increases.

As this algorithm comprises a significant number of distinct decision trees, subsamples of the initial training data are generated using the bootstrapping procedure, and then new decision trees are generated using these subsamples. From each of the decision trees, a random subset of the features is selected while completing the training. Finally, for better accuracy, the average of all the results of these decision trees will be computed and this is known as aggregation. Bootstrap ensures that the same data is not used more than once which makes the model to be less sensitive to the main origin data and this random selection feature reduces the correlation between the decision trees which will cause less variance between the trees. Even though the algorithm requires more processing time than standard decision trees, the model still guarantees model diversification.

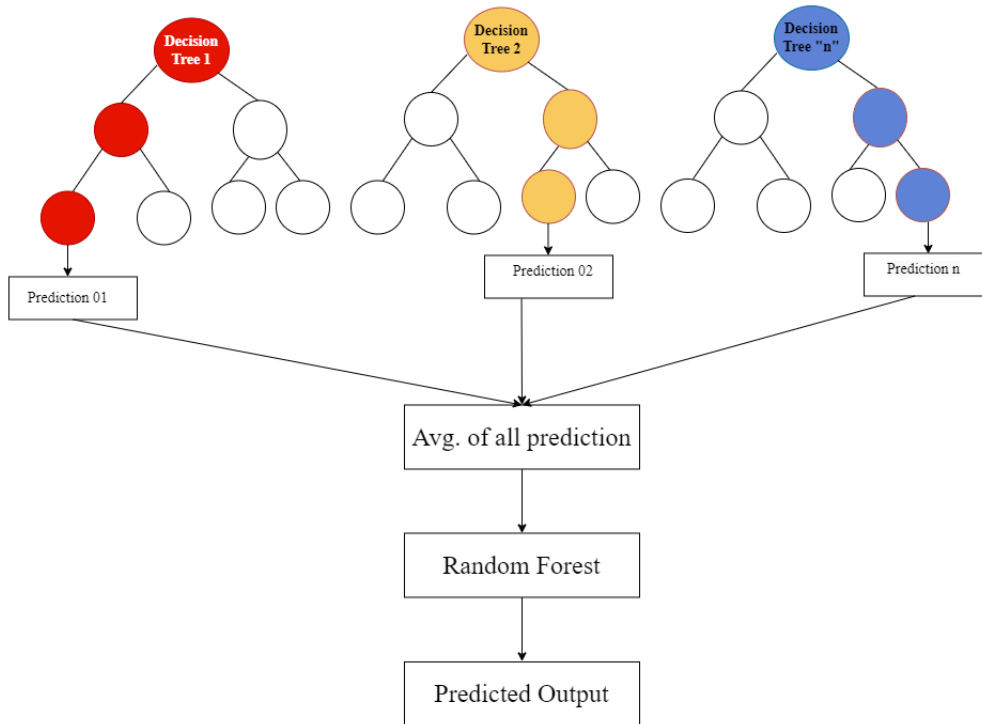


Figure 2.7: Random forest classifier Tree

## Logistic Regression

Logistic Regression is an algorithm for machine learning that is a predictive analytic model based on the concept of probability. This algorithm is used to solve classification problems in order to distinguish between classes. In the logistic regression procedure, there are three class categories. These are binary, multi-class, and ordinal classes. Here, binary classification is the simplest example, with the only possible values being Yes and No or True and False. Logistic regression is used to employ a categorical variable as a dependent variable. To classify whether a condition is infectious or not, for instance, the logistic regression approach is superior to linear regression because it provides more accurate results for categorical variables. Finding the likelihood that a variable will occur is the goal of the logistic regression technique. There is no space for error because the range of the predicted values is restricted to 0 to 1. Using the Sigmoid Function, which converts a real number to a number between 0 and 1, it is possible to accomplish the objective. This function's formula is 2.8:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

This dependent value,  $f(x)$ , will range between 0 and 1 and is between 0 and 1. As well as the independent variable  $x$  is the input variable. This function transforms an

independent variable into a probability expression between 0 and 1 with regard to the dependent variable. The following diagram 2.8 illustrates the sigmoid function:

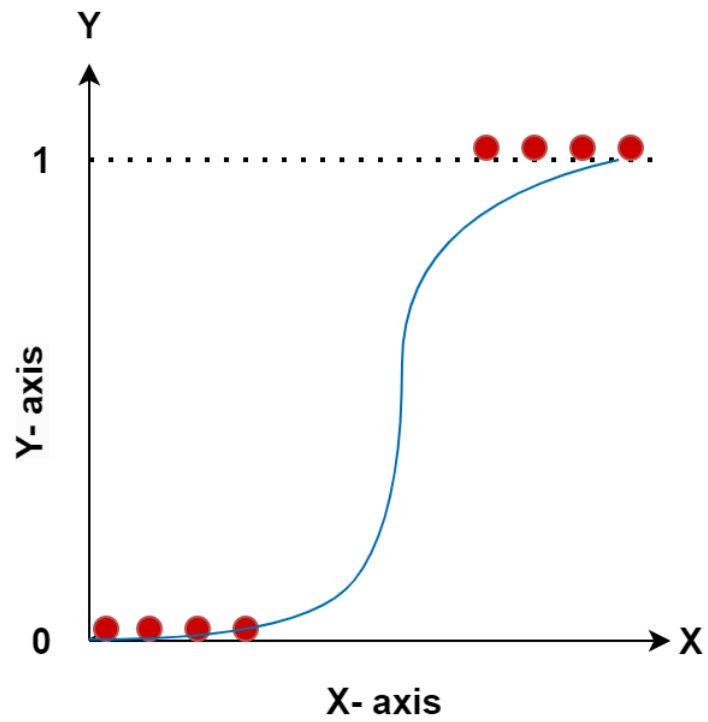


Figure 2.8: logistic regression plot



# Chapter 3

## Methodology

### 3.1 Research Approach

This research methodology 3.1 involves collecting data, pre-processing it (including negative image, data augmentation, rescaling and resizing, and encoding), selecting pre-trained models for feature extraction (ResNet-50, VGG16, and Inception-v3), splitting the dataset into train-test groups, constructing a classification-focused CNN and MLP architecture, training the CNN classifier with the extracted features, and classifying AbdomenCT, headCT, BreastMRI, ChestCT, Hand, and CXR using the CNN classifier to analyze accuracy. Furthermore, federated learning is implemented to interpret the model's prediction and make healthcare data more secure by avoiding data transfer and transmitting models. Finally, SVM, Logistic Regression, and Random Forest classification are applied to analyze the performance of the CNN model and compare it to other pre-trained models as feature extractors on our new hybrid dataset.

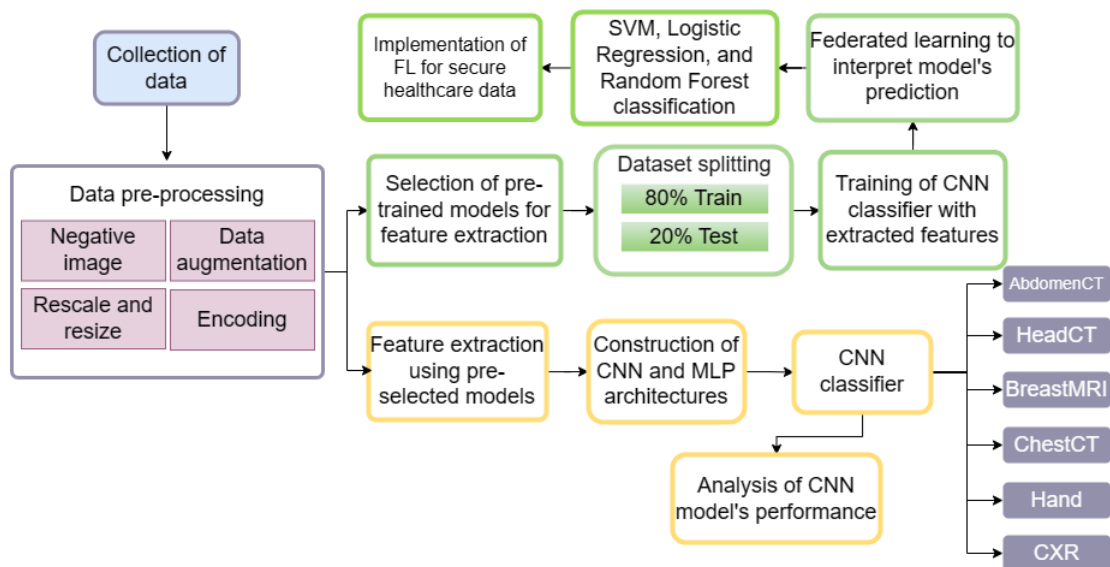


Figure 3.1: Methodology Model

## 3.2 Data Pre-processing

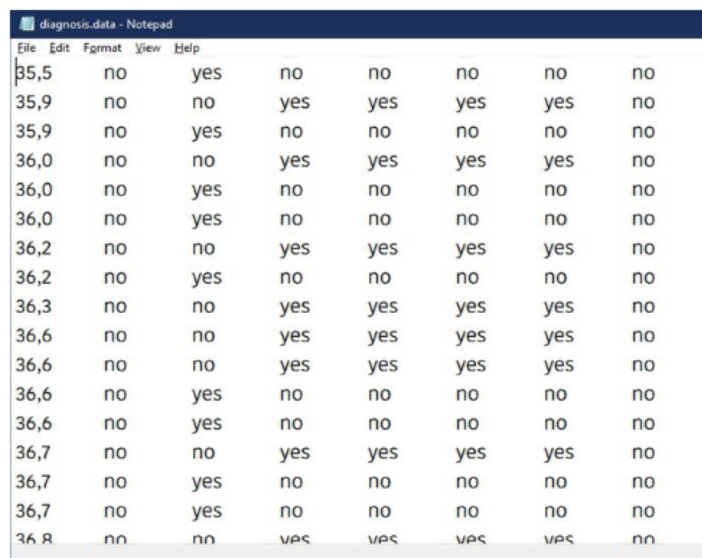
### 3.2.1 Dataset

A composite dataset of colored images that was compiled from several sources, including Medical MNIST, was used in this study’s training of the models. [17] 10,000 of them are associated with the HeadCT, 10,000 are involved with the AbdomenCT, 8954 are related with the BreastMRI, 10,000 are linked with the ChestCT, 10,000 belong to the Hand, and the rest of the 10,000 images belong to CXR.

```
#unzip train data there,  
! unzip /content/medical-mnist.zip -d train  
Streaming output truncated to the last 5000 lines.  
inflating: train/HeadCT/005000.jpeg  
inflating: train/HeadCT/005001.jpeg  
inflating: train/HeadCT/005002.jpeg  
inflating: train/HeadCT/005003.jpeg  
inflating: train/HeadCT/005004.jpeg  
inflating: train/HeadCT/005005.jpeg  
inflating: train/HeadCT/005006.jpeg  
inflating: train/HeadCT/005007.jpeg  
inflating: train/HeadCT/005008.jpeg  
inflating: train/HeadCT/005009.jpeg  
inflating: train/HeadCT/005010.jpeg  
inflating: train/HeadCT/005011.jpeg  
inflating: train/HeadCT/005012.jpeg  
inflating: train/HeadCT/005013.jpeg  
inflating: train/HeadCT/005014.jpeg  
inflating: train/HeadCT/005015.jpeg  
inflating: train/HeadCT/005016.jpeg
```

Figure 3.2: Representation of dataset1

A further dataset from “UCT” called MEDICAL DIAGNOSIS DATA [41] is used in this study. This information includes the patient’s body temperature, any obvious symptoms of nausea, back pain, pushing during urination, burning or itching in the urethra, swelling at the urethra outlet, bladder inflammation, and nephritis with renal pelvic origin. The patient’s temperature fluctuates a lot, between 35 and 42 degrees Celsius. The remaining information consists of only yes and no. 3.3



Temperature	nausea	back pain	pushing during urination	burning or itching in the urethra	swelling at the urethra outlet	bladder inflammation	nephritis with renal pelvic origin
35,5	no	yes	no	no	no	no	no
35,9	no	no	yes	yes	yes	yes	no
35,9	no	yes	no	no	no	no	no
36,0	no	no	yes	yes	yes	yes	no
36,0	no	yes	no	no	no	no	no
36,0	no	yes	no	no	no	no	no
36,2	no	no	yes	yes	yes	yes	no
36,2	no	yes	no	no	no	no	no
36,3	no	no	yes	yes	yes	yes	no
36,6	no	no	yes	yes	yes	yes	no
36,6	no	no	yes	yes	yes	yes	no
36,6	no	yes	no	no	no	no	no
36,6	no	yes	no	no	no	no	no
36,7	no	no	yes	yes	yes	yes	no
36,7	no	yes	no	no	no	no	no
36,7	no	yes	no	no	no	no	no
36,8	no	no	yes	yes	yes	yes	no

Figure 3.3: Representation of dataset2

### 3.2.2 Dataset Collection

This image 3.4 is a representation of the Medical MNIST dataset.[17] These photos are discovered after being categorized into 6 classes using CNN and MLP models. Rgb data from the gathered data are displayed in the first image.

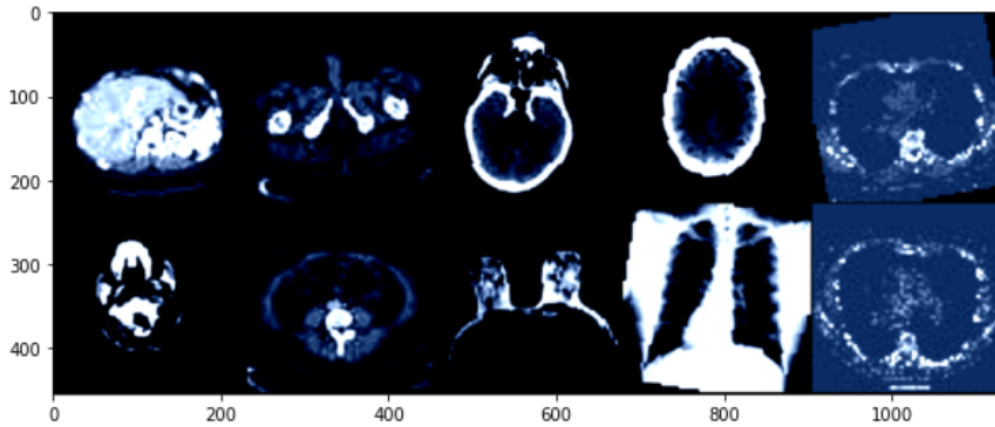


Figure 3.4: Dataset Collection

The second image 3.5 shows a situation that is fairly similar to the first. However, as it lacks a RGB value, the image is displayed in black and white.

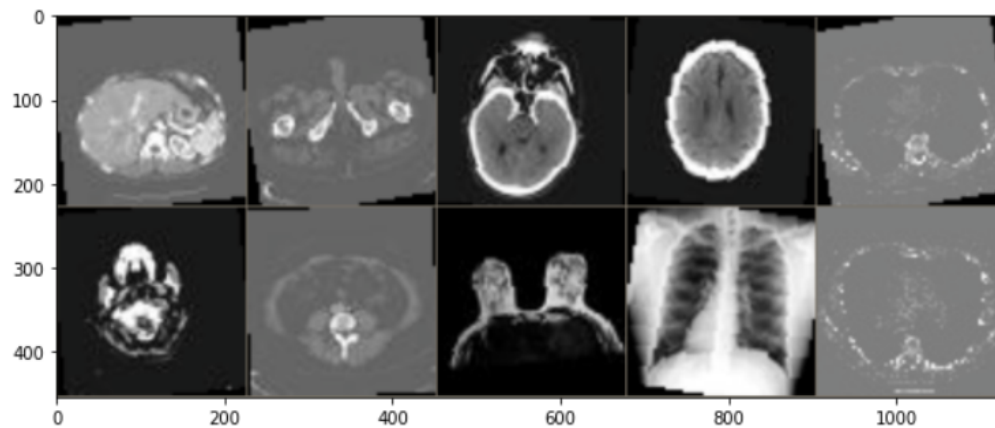


Figure 3.5: Dataset Collection

### 3.2.3 Rescale And Resize

The image pixel values have also been rescaled between 0 and 1 to standardize the data prior to feature extraction. Generally, an image's pixel values range from 0 to 255. Supervised learning is not suited to this wider range of pixels because it causes destabilization in the neural network and makes it challenging for the model to handle such enormous numbers. Therefore, the pixels must be rescaled before being given into the model for the best outcome. In order to resize the pixels, they have been split by 255.

The hybrid dataset utilized in this experiment was generated from a collection of images of multiple lengths. The input images in this study work have been scaled in line with the specified size of the input of the pre-trained techniques. Since these have been employed for feature extraction. Prior to being input into the summarizes for the VGG16 and ResNet50V2 models, all of the photos were downsized to 224x224 pixels; however, for the InceptionV3 and InceptionRes-NetV2 models, this was adjusted to 150x150 pixels and 299x299 pixels, correspondingly.

The processing of high-definition images can be lengthy and complicated, which is an additional factor why the images were reduced in size before being inputted.

### 3.2.4 Encoding

Data transformation into a structure that can be used by most technologies or by any independent process is the primary objective of encoding. It is inadequate for data security because encoding is done using a variety of publicly accessible techniques. In this hybrid dataset, there are six distinct classifications of images, thus the image labels have only performed one hot encoding. To do this, the feature extraction procedure's class mechanism was changed to "categorical", resulting in a 2D NumPy array with one hot-encoded description.

### 3.2.5 Data Augmentation

The process of artificially generating additional data from training examples already present is defined as data augmentation. Resizing, inverting, zooming and scaling, clipping, padding, and other approaches are among them. This optimizes the model's performance by making it more resilient while developing solutions like overfitting and data limitation.

## 3.3 Federated Learning

The idea of federated learning has just been offered by internet giant Google. Google's main objective is to create machine learning models using datasets distributed across many devices while preventing data theft. So, The approach is based on how it can establish data security for e-health records using federated learning which is focused on the "FedAvg" Federated averaging technique.

The earliest basic Federated learning algorithm (equation 3.1 and 3.2) invented by Google for Federated learning challenges is called FedAvg. Till now, a series of FedAvg algorithms have been designed, notably "FedProx," "FedMax," "FedOpt," and some others, to handle several issues with Federated learning. FedAvg seeks to maximize the objective of the global model, which is simply the weighted average of the loss of the regional device combined cumulatively for each round.

$$f(w) = \sum_{k=1}^k \frac{\eta^k}{\eta} F_k(w) \quad (3.1)$$

where,

$$F(k) = \frac{1}{\eta_k} \sum_{i \in P_k} f_i(w) \quad (3.2)$$

A strategy involving the collection of the data set, the training of the FedAvg, FedMax, and FedSVRG models, and the evaluation of their accuracy is suggested. In order to evaluate the model accurately, the dataset was split into training and test sets. To remain devoted to the original study, it has been decided to adhere to the predetermined training and administer the tests in accordance with the methods outlined in the study. The obtained dataset was split around 8:2 for the train and test set for the training, testing, and extraction in order to run the models on them and obtain the required results.

### 3.4 Proposed Model

Federated learning is now a well-liked and secure algorithm. After CNN and MLP models have been used to train datasets, FL is used to aggregate, analyze, and improve data accuracy. Federated learning allows several IoT devices to learn a model concurrently without exchanging data. In this study, a federated learning algorithm was utilized to improve data security. On this dataset, the FedAVG, FedMAX, and FedSVRG methods have been applied. This will compare models to decide which algorithm provides the highest level of security. In this situation, the cloud model, which serves as the central model, is utilized. Each user is provided with a model from the cloud model. These models utilize user data to generate a new user model. New user models have improved the precision of data analysis. Eventually, updated user models broadcast back to the cloud to update the central model. Since no data is transferred, federated learning is more effective and secure than other algorithms.

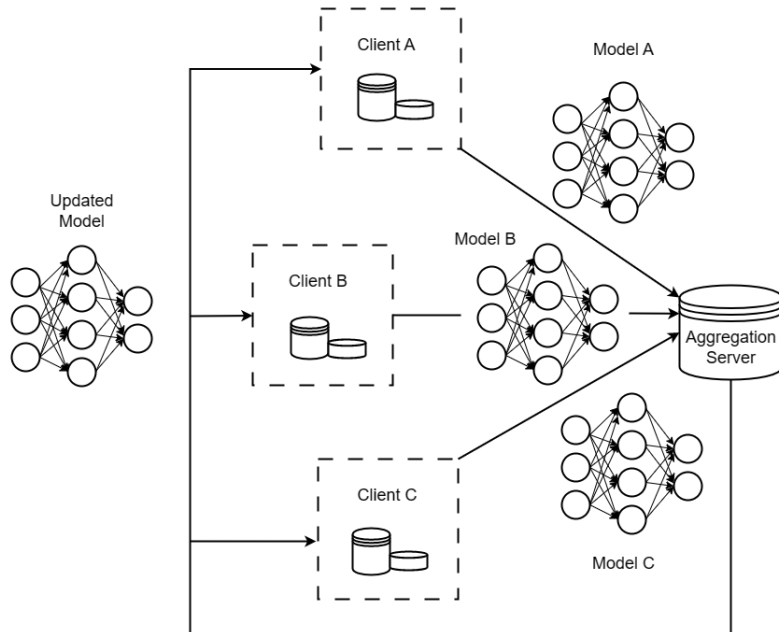


Figure 3.6: Federated Learning (FL) model.

The suggested model 3.6 is structured around the five essential steps listed below:

1. A randomly chosen sample of customers or devices is gathered.
2. Also every user gets broadcasting of the server’s developed model.
3. Stochastic Gradient Descent (SGD) is applied in parallel by the clients to their own loss functions, and the resulting model is then sent to the server for aggregation.
4. The server eventually averages those local models to update the machine learning model.
5. After that, the method 3.7 is repeated for n consecutive transmission cycles.

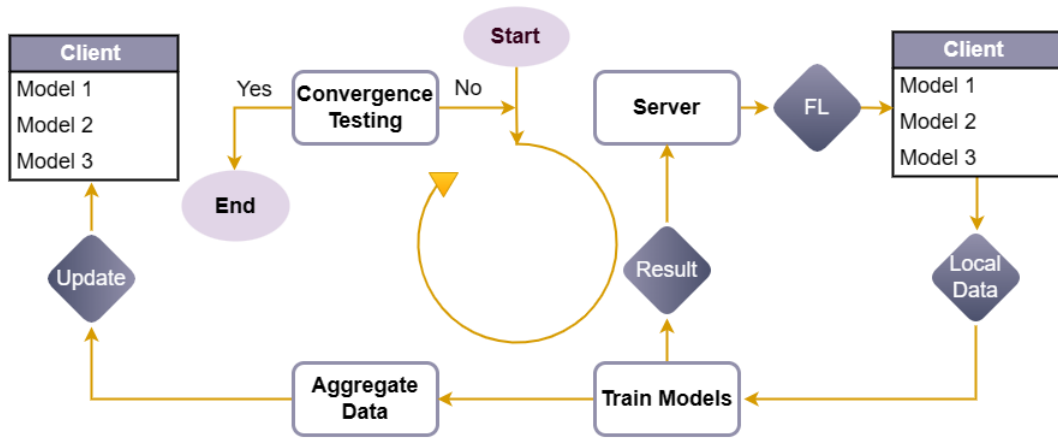


Figure 3.7: Working Mechanism of FL

### 3.4.1 Stochastic Gradient Descent

The stochastic gradient descent algorithm is employed to determine the minimum of a function. It is an iterative approach in which the gradient of the function is approximated with a randomized data point at each step. The method then proceeds in the direction of the gradient to identify the function’s minimum.

A process or system associated with a random probability is termed stochastic. Only a few samples, rather than the complete data set, are randomly picked for each iteration of stochastic gradient descent. A dataset’s sample count used to calculate the gradient for each iteration of the Gradient Descent algorithm is referred to as a “batch.” The batch in traditional Gradient Descent optimization methods, like Batch Gradient Descent, represents the complete dataset. Even while using the complete dataset makes it easier to find minima in a less noisy and random fashion, the issue arises when the dataset is too big.

Each iteration of Gradient Descent will require the utilization of all samples if users employ a traditional Gradient Descent optimization technique and your dataset has one million samples. Until the minimum quantity is obtained, this process must be

repeated. As a result, doing so is extremely computationally expensive. A single sample, or batch size of 1, is used for each SGD iteration. The sample is chosen and rearranged at random to complete the loop.

Batch Gradient Descent follows the route 3.8 indicated below:

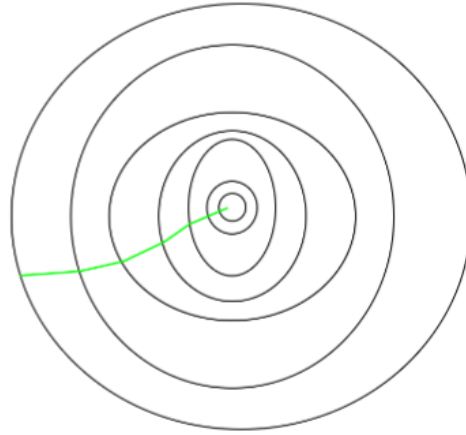


Figure 3.8: Stochastic Gradient Descent has gone down a certain route.

### 3.4.2 SGD Algorithm

An approach of algorithm 1 to clustering a set of data points into  $k$  groups is the  $k$ -means algorithm. The process begins by randomly choosing  $k$  centroids, after which it allocates each data point to the cluster that contains the closest centroid. The system next calculates the average of the data points in each cluster and changes the centroid position to this average. Repeating this cycle will stop the centroids from shifting. The distance between data points and the cluster center must be kept to a minimum.

---

#### Algorithm 1 Simplified Density-Based Clustering Algorithm (SDG)

---

A set of data points A set of fitted parameters Randomly select  $k$  centroids

**for** each data point **do** Calculate the distance to the nearest centroid Assign the data point to the centroid with the smallest distance

**for** each centroid **do** Calculate the mean of the assigned data points Update the centroid with the calculated mean

**if** centroids remain unchanged **then** Stop

**else** Go back to step 1

---

### 3.4.3 FedAVG

The algorithm 2 FedAVG, often referred to as Federated Averaging, creates the aggregated model by taking the weighted average of all model updates.[37] A communication efficient approach for distributed training with a sizable client base is federated averaging (FedAvg). Device A, for instance, sends model A with the value 0.6 to the server. For example, device B transmits model B to the server with a

value of 0.1. Device C transmits model C to the server with a value of 0.2. The server calculates these three numbers as follows:  $(0.6 + 0.1 + 0.2)/3 = 0.3$   
The server provides A, B, and C. An aggregated model with the value of 0.3.

The Federated Averaging technique combines them together to provide a more precise model. This concept comes from Google’s “productionisable” feature. In the subject of Federated Learning, this algorithm is crucial.

---

**Algorithm 2** Federated Averaging

---

```

1: Initialize  $\omega_0$ 
2: for each round  $t = 1, 2, \dots$  do
3:    $m \leftarrow \max(C.K, 1)$ 
4:    $S_t \leftarrow$  (random set of  $m$  clients)
5:   for each client  $k \in S_t$  in parallel do
6:      $\omega_{k,t+1} \leftarrow \text{ClientUpdate}(k, \omega_t)$ 
7:   end for
8:    $\omega_{t+1} \leftarrow \sum_{k=1}^K \eta_k \eta \omega_{k,t+1}$ 
9: end for

```

---

### 3.4.4 FedMAX

FedMAX is one of the conventional algorithm in federated learning.[26] This algorithm 3 can handle dispersed non-identical data. It is particularly effective at communicating data since it uses weights in its operation. Learning models require a variety of data produced by IoT devices in order to create smarter apps. A promising privacy-preserving learning technique that separates model training from the need for access to personal data is federated learning. FedMAX is a highly reliable and effective distributed federated learning technology to address these problems.

---

**Algorithm 3** FedMax Algorithm

---

```

1: Input: Local datasets  $D_1, \dots, D_n$ 
2: for  $i \leftarrow 1, \dots, n$  do
3:   Randomly sample  $D_i$  to obtain  $D'_i$ 
4:   Calculate a local model  $M_i$  on  $D'_i$ 
5: end for
6: Output: Global model  $M$ 
7: Calculate a global model  $M$  on  $\{D'_1, \dots, D'_n\}$ 

```

---

### 3.4.5 FedSVRG

FedSVRG Based Federated Learning in MEC Networks Communication is an Efficient Scheme. This algorithm 4 reduces the cost of user-MEC server communication.[37] FedSVRG provides greater precision and converges more quickly, which



translates to less communication expense. In FedSVRG, there are two iteration loops. Convergence rates will be increased by using the average gradient rather than updating it each time during the inner loop rounds for a predetermined number of iterations. As a result, the number of iterations between the participants and the MEC server may be significantly reduced.

---

**Algorithm 4** Federated Stochastic Variance Reduced Gradient (FedSVRG)

---

**Require:**  $K$  - number of global rounds

$N$  - number of clients

$M$  - number of local rounds

$\theta_0$  - initial global parameters

$\mathcal{D}_i$  - local dataset of client  $i$

**for**  $k = 1, \dots, K$  **do**

    Calculate  $\nabla f_{\text{avg}}(\theta_k)$  by averaging gradients from each client

**for**  $i = 1, \dots, N$  **do**

        Client  $i$  computes  $\theta_k^i \leftarrow \theta_k - \frac{1}{M} \sum_{t=1}^M \nabla f_i(\theta_k^i) - \frac{1}{M} \sum_{t=1}^M \nabla f_i(\theta_k^i; x_{i,t}, y_{i,t})$   
        with local dataset  $\mathcal{D}_i = \{(x_{i,t}, y_{i,t})\}_{t=1}^M$

**end for**

$\theta_{k+1} \leftarrow \theta_k - \eta \nabla f_{\text{avg}}(\theta_k)$

**end for**

---

# Chapter 4

## Implementation and Result

### 4.1 Import data from Kaggle

Kaggle libraries were first installed. The Kaggle API is required to access the website. The Kaggle user profile contains the Kaggle API. Then, utilizing that API, any data may be imported; all that is required is the username and dataset name. Data is often downloaded as a zip file. So, the Kaggle data that was downloaded (fig: 4.1 ) required to be unzipped.[17]

```
# First, we'll need to authenticate with Kaggle in order to access the data
!pip install kaggle
!mkdir .kaggle

from google.colab import files
files.upload()

!rm -r ~/.kaggle
!mkdir ~/.kaggle
!mv ./kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets list

# Next, we'll download the data from Kaggle
!kaggle datasets download -d andrewmvd/medical-mnist

# Unzip the data and move it to the appropriate directory
!unzip /content/medical-mnist -d data
```

Figure 4.1: Import data from Kaggle

### 4.2 Using the CNN classifier, analyze the performance of the different pre-trained models

#### 4.2.1 ResNet50

The training accuracy was determined to be 96.97% and the validation accuracy was determined to be 96.47% during pre-processing in feature extraction throughout 5 epochs.

Text(0, 0.5, 'accuracy')

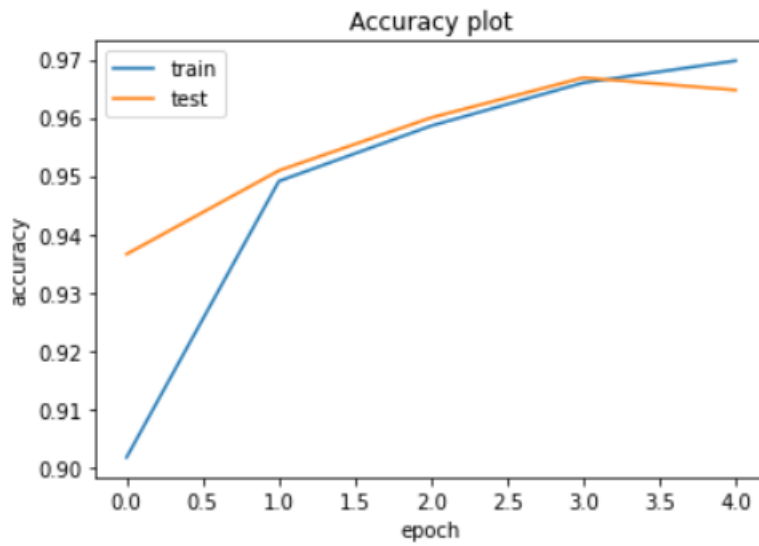


Figure 4.2: ResNet50 Validation Accuracy Plot

The accuracy plot of ResNet50 is displayed in figure 4.2. The validation accuracy increases marginally from the first to the tenth epoch, whereas the training accuracy rose significantly. After that, the accuracy grows simultaneously from the tenth to the thirty-first epoch. The training accuracy rises in the following section, from the thirty-fifth to the forty-fifth, while the test accuracy starts to fall.

## 4.2.2 VGG16

During pre-processing in feature extraction over 5 epochs, the accuracy for training was found to be 99.84%, and the accuracy for validation was found to be 99.81%.

Text(0, 0.5, 'accuracy')

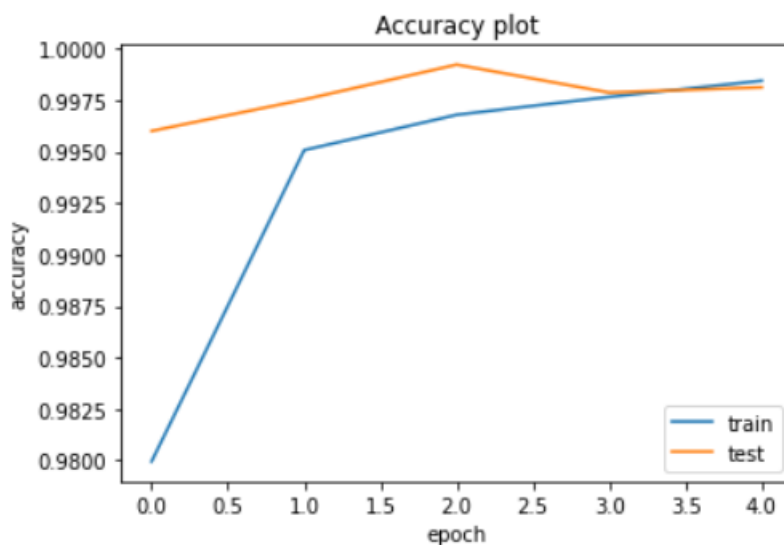


Figure 4.3: VGG16 Validation Accuracy Plot

Figure 4.3 displays the accuracy of the VGG16 in various epochs. The validation accuracy increases to its maximum between the first and twentieth accuracy epochs, after which it starts to decline until the thirty-first accuracy epoch. After that, the test accuracy remains constant. In contrast, the training accuracy dramatically increases from the first to the tenth epoch, then modestly increases from the tenth to the twentieth epoch and remains stable.

### 4.2.3 InceptionV3

Preprocessing in feature extraction over 5 epochs revealed accuracy for training to be 99.90% and accuracy for validation to be 99.96%.

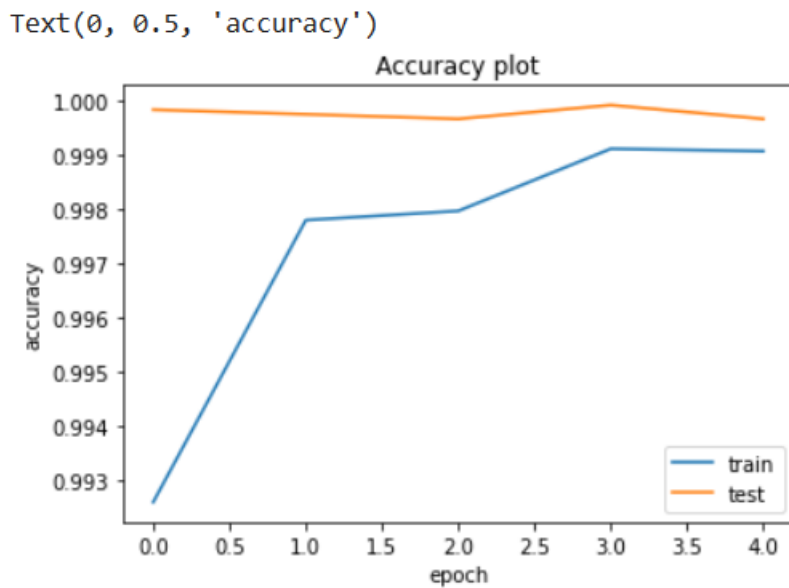


Figure 4.4: InceptionV3 Validation Accuracy Plot

Figure 4.4 displays the InceptionV3 accuracy plot. Every epoch has very poor validation accuracy. In the twentieth epoch, it slightly declines, while in the thirty first epoch, it slightly grows. Then, it resumes a small decline. On the other hand, up to the ninth epoch, accuracy greatly improves. The accuracy then gradually increases until the twentieth, climbs again until the thirty first, and then remains almost constant after that.

After extracting the features using base models, 80% of the collected features were preserved in the training set while 20% were preserved in the validation set after the retrieved features were split into train and validation sets. Following that, the CNN classifier was trained using the training data, and the validation data was used to assess its performance. To comprehend the performance in greater detail, the same batch size and different amounts of epochs were used to train each of the three models. In the following table, the validation accuracy of each of the three models for varying numbers of epochs is illustrated.

Accuracy			
Base Model	5 epochs	10 epochs	20 epochs
VGG16	99.81	99.93	99.94
ResNet50	96.47	97.19	98.00
InceptionV3	99.96	99.96	99.97

Table 4.1: Validation accuracy for different epoch counts.

Table 4.1 demonstrates how the number of accuracy increases along with the number of epochs. However, it starts to fluctuate if we overtrain. In this case, InceptionV3 displays the highest accurate result, which is 99.97% in 20 epochs. VGG16 and ResNet50 follow with 99.94% and 98%, respectively. Thus, InceptionV3, VGG16, and ResNet50 are in order of best accuracy.

#### 4.2.4 Comparison of the accuracy loss plot for (i) VGG16, (ii) ResNet50, and (iii) InceptionV3

Text(0, 0.5, 'loss')

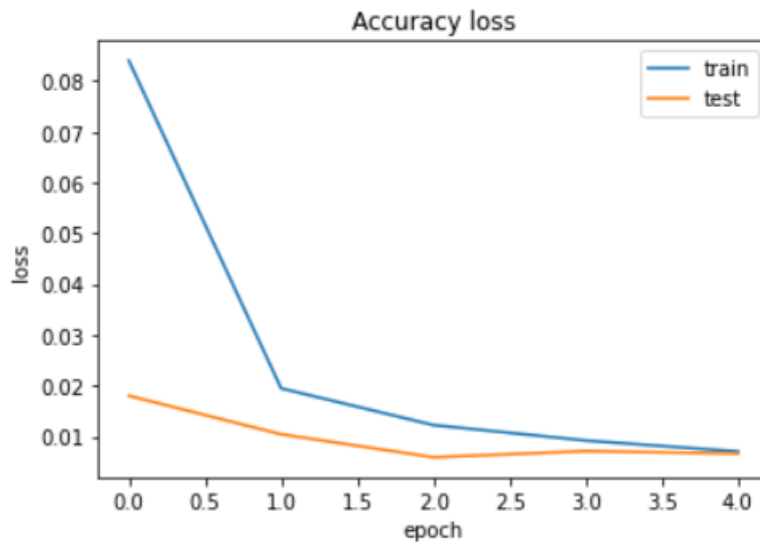


Figure 4.5: Accuracy loss plot for (i) VGG16

Text(0, 0.5, 'loss')

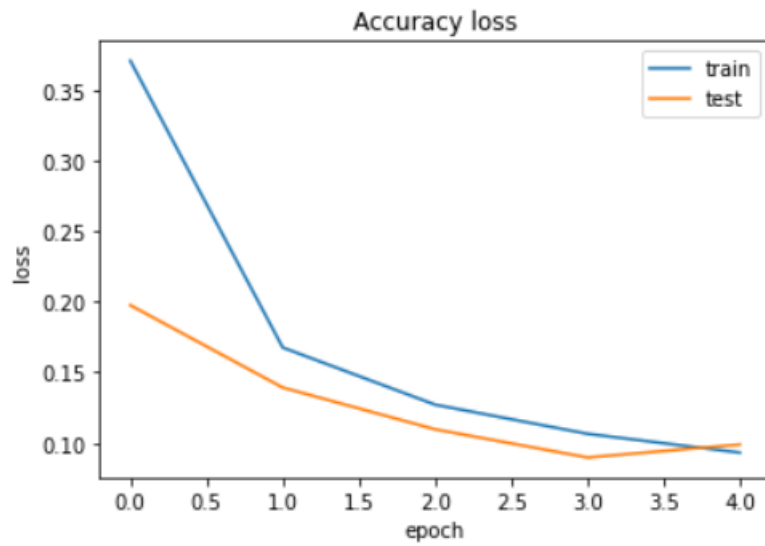


Figure 4.6: Accuracy loss plot for (ii) ResNet50

Text(0, 0.5, 'loss')

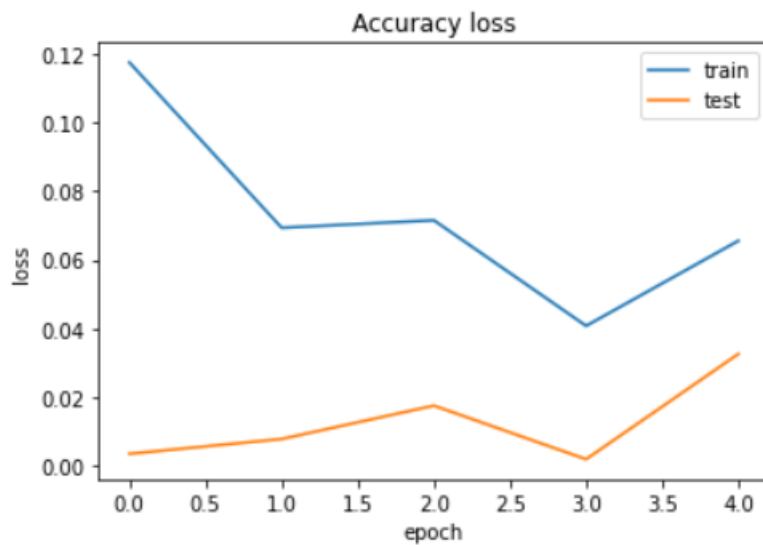


Figure 4.7: Accuracy loss plot for (iii) InceptionV3

The accuracy loss and validity loss are depicted in figures 4.5, 4.6, 4.7. The training loss dramatically lowers in the first epoch for VGG16, ResNet50, and InceptionV3. After the first epoch, the training loss gradually dropped for VGG16 and ResNet50, but started to increase for InceptionV3. In addition, whereas InceptionV3 swings, VGG16 and ResNet50 decrease during the course of each epoch. It declines in the second and third epochs before increasing once more in the fourth epoch. In the validation epoch, the scenario is nearly the same, with the exception that ResNet50 experiences a rise in accuracy loss while InceptionV3 experiences an almost exact reverse.

## 4.2.5 Analysis of the three feature extractors' performance using the random forest classifier

After applying VGG, ResNet and InceptionV3 three base models, random forest classifier has been implemented.

Random Forest%			
Feature extractors	5 epochs	10 epochs	20 epochs
VGG	91.11	95.55	95.55
ResNet	91.11	97.77	95.55
InceptionV3	95.55	97.77	97.77

Table 4.2: Validation accuracy for different Random Forest.

InceptionV3, ResNet50, and VGG16 accuracy all remain constant after ten epochs, according to the random forest table 4.2. It can be shown that InceptionV3 exhibits the highest degree of accuracy. The accuracy is then the same for VGG16 and ResNet50. But up until the eleventh epoch, VGG16 and ResNet50 provide distinct outcomes. Here, we may obtain InceptionV3 as the most accurate source.

### MLP And CNN Models

While both a multi-layer perceptron (MLP) and a convolutional neural network (CNN) are classes of neural networks, their applications and architectural designs differ.

Three completely connected layers (in figure 4.8) make up this straightforward multi-layer perceptron model. The first layer has 200 neurons, and the input shape is described as "shape." Except for the final layer, which uses a softmax activation function, all layers employ ReLU as their activation function. The model is made for classification problems and has "classes" neurons in the top layer.

```
class SimpleMLP:
    @staticmethod
    def build(shape, classes):
        model = Sequential()
        model.add(Dense(200, input_shape=(shape,)))
        model.add(Activation("relu"))
        model.add(Dense(200))
        model.add(Activation("relu"))
        model.add(Dense(classes))
        model.add(Activation("softmax"))
        return model
```

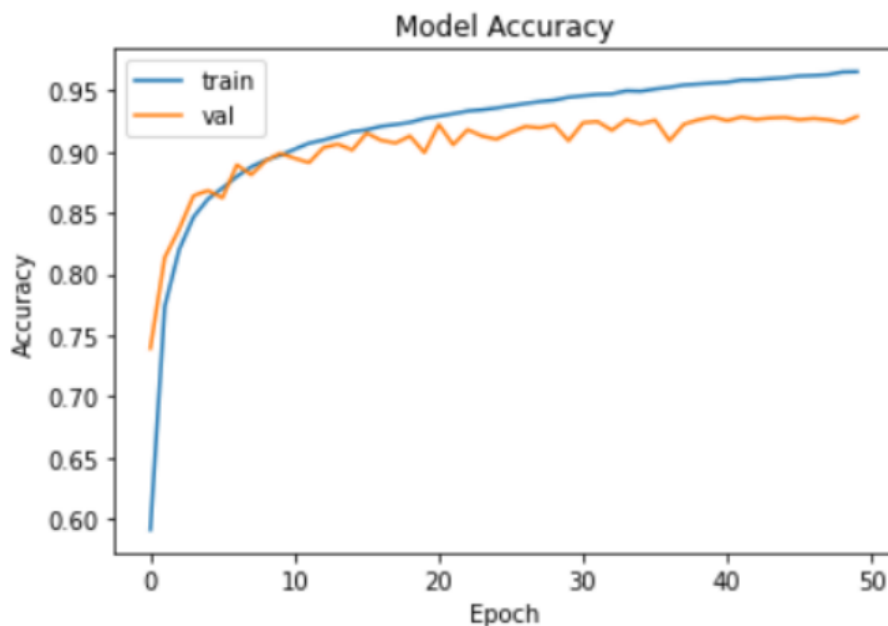
Figure 4.8: Simple MLP

This straightforward model 4.9 contains two convolutional layers with, respectively, 32 and 64 filters and kernel sizes of (3,3). The shape parameter specifies the model's input shape. With the exception of the top layer, which utilizes the softmax activation function, all layers use the ReLU activation function. Additionally, the model

employs max-pooling layers with a pool size of (2,2) to lessen the input's dimensionality. The model's last layer, which is intended for image classification tasks, has "classes" neurons, or the number of classes in the dataset.

```
class SimpleCNN:
    @staticmethod
    def build(shape, classes):
        model = Sequential()
        model.add(Conv2D(32, (3,3), input_shape=shape))
        model.add(Activation("relu"))
        model.add(MaxPooling2D(2,2))
        model.add(Conv2D(64, (3,3)))
        model.add(Activation("relu"))
        model.add(MaxPooling2D(2,2))
        model.add(Flatten())
        model.add(Dense(256))
        model.add(Activation("relu"))
        model.add(Dense(classes))
        model.add(Activation("softmax"))
        return model
```

Figure 4.9: Simple CNN



Test loss: 0.0932680070400238  
Test accuracy: 0.9722222089767456

Figure 4.10: CNN Accuracy Graph

The CNN was implemented following pre-processing with various feature extraction models using machine learning classifiers. The figure 4.10 illustrates how training accuracy grows noticeably up until the tenth epoch. The training accuracy is very



steady after the tenth period. On the other hand, the validation accuracy starts off considerably higher before varying across each epoch. However, the table shows that despite the significant fluctuation, it still follows a relatively predictable pattern.

## 4.3 Implementation of Federated learning Algorithm

### 4.3.1 Creating Clients

This function uses a provided label list and picture list to create clients for a federated learning system. It asks for the desired client count as well as the client’s first name. The data is divided into “shards” and made random before being distributed to each client. The data shards (tuples of picture and label lists) are returned as a dictionary with the client names as the keys and the values. The number of clients and the number of data shards must match.

```
def create_clients(image_list, label_list, num_clients=10, initial='clients'):
    ''' return: a dictionary with keys clients' names and value as
        data shards - tuple of images and label lists.
    args:
        image_list: a list of numpy arrays of training images
        label_list: a list of binarized labels for each image
        num_client: number of fedrated members (clients)
        initials: the clients' name prefix, e.g, clients_1
    ...

    #create a list of client names
    client_names = ['{}_{}'.format(initial, i+1) for i in range(num_clients)]

    #randomize the data
    data = list(zip(image_list, label_list))
    random.shuffle(data)

    #shard data and place at each client
    size = len(data)//num_clients
    shards = [data[i:i + size] for i in range(0, size*num_clients, size)]

    #number of clients must equal number of shards
    assert(len(shards) == len(client_names))
```

Figure 4.11: Creating Clients

### Model Learning Rate

The learning rate for federated learning is the fundamental task of this model after creating clients. Lowering the learning rate will make FL run more speedily. Additionally, it features a round of 11 that specifies the number of times the federated learning model will run and the learning rate is 0.01. Also, the momentum is 0.9.

### 4.3.2 Aggregation through Federated Averaging

The code specifies the weight scaling factor, scale model weights, sum scaled weights, and test model four federated learning routines. The scaling factor for the client’s model weights is returned by the weight scaling factor function, which accepts a dictionary of clients’ training data and a client name. When a model’s weights and a scaling factor are sent in, the scale model weights method returns the scaled weights.

The sum scaled weights function accepts a list of scaled weights and outputs the weights' average or the sum of the scaled weights. The test model function evaluates the accuracy and loss of the model on the test data and prints it out. It accepts test data, a model, and a communication round.

```
def weight_scaling_factor(clients_trn_data, client_name):
    client_names = list(clients_trn_data.keys())
    #get the bs
    bs = list(clients_trn_data[client_name])[0][0].shape[0]
    #first calculate the total training data points across clients
    global_count = sum([tf.data.experimental.cardinality(clients_trn_data[client_name]).numpy() for client_name in client_names])*bs
    # get the total number of data points held by a client
    local_count = tf.data.experimental.cardinality(clients_trn_data[client_name]).numpy()*bs
    return local_count/global_count

def scale_model_weights(weight, scalar):
    '''function for scaling a models weights'''
    weight_final = []
    steps = len(weight)
    for i in range(steps):
        weight_final.append(scalar * weight[i])
    return weight_final
```

Figure 4.12: Aggregation through Federated Averaging

## 4.4 Result

Models	Accuracy	Log Loss
CNN with FL	99.661%	1.0511
CNN without FL	97.22%	0.0912
MLP with FL	98.592%	1.0724
MLP without FL	97.83%	0.0933

Table 4.3: Summary of Results

From the table 4.3, we may conclude that FL offers greater accuracy. According to the table, CNN has an accuracy rating of 97.22% without FL and 99.66% with FL. If we examine the lag loss, CNN with FL provides a little bit greater loss than CNN without FL. On the other hand, the MLP scenario is quite similar. Additionally, MLP with FL offers superior accuracy (98.592%) than MLP without FL (97.83%), while MLP with FL's lag loss is marginally higher.

### 4.4.1 Previous Results

The average validation accuracy rate for the MLP model in [table 4.4] is 93.81% for IID, and the average validation accuracy rate is 72.31% for Non-IID. However, using the CNN model, the average validation accuracy rate is 97.29% for IID, while the average validation accuracy rate is 79.82% in Non-IID. Using IID and Non-IID data, both models have been investigated. IID data enhances overall accuracy.

Model	Acc. of IID	Acc. of non-IID
FedAVG-CNN	96.29%	79.82%
FedAVG-MLP	93.81%	72.31%

Table 4.4: Results of 10 epochs training with the learning rate of 0.01.

Model	Acc. of IID	Acc. of non-IID
FedAVG-CNN	98.89%	92.61%
FedAVG-MLP	96.31%	91.03%

Table 4.5: Results of 50 epochs training with the learning rate of 0.01.

Additionally, it is found that accuracy in 50 epochs [table 4.5] was significantly higher than in 10 epochs. The average validation accuracy for MLP in IID and Non-IID is 95.31% and 91.03%, respectively. In comparison, CNN’s validation accuracy is 97.89% and 92.61%. As more training epochs are used, the model gets more precise.

#### 4.4.2 Comparison with related works

In table 4.6, the first paper works with the Covid-19 dataset. Only FL was used in this research and they found Local accuracy 79.5% and FL Global accuracy 92%. After that, In the second the paper works with a dataset consisting of 3000 sample images of lichen planus, acne and SJS ten. In their research they used CNN method and they ended up having a global accuracy of 96%. In the third research paper they used CDS, FL, CIIL. In the paper CDS global accuracy = 86.2%, FL global accuracy = 85.7%, CLL global accuracy = 85.3%. Lastly, in our research, the medical MNIST dataset is being used. For Feature extraction resNet50, VGG16 and Inception-v3 is used. After that CNN and MLP model have been implemented individually for further accuracy. In the last, the FL hybrid model was used to find the best accuracy. CNN with FL provided 99.661% accuracy whereas CNN without FL provided 97.22% accuracy. On the other hand, MLP with FL provided 98.592% accuracy and MLP without FL provided 97.83% accuracy.

It is clear through study and comparisons with other articles that the papers used various methodologies and datasets. However, it is clear from our research that utilizing a hybrid federated learning model yields the best accuracy of 99.661%.

Comparison Table					
Sl no	Topic	Author	Dataset used	Method	Performance
1	Federated learning for predicting clinical outcomes in patients with COVID-19[31]	I Dayan, HR Roth, A Zhong, A Harouni, A Gentili	COVID-19 data	FL	Local Acc: 79.5% FL global Acc: 92%
2	Machine Learning Algorithms based Skin Disease Detection[10]	Bhadula, S., Sharma, S., Juyal, P., Kulshrestha, C. (2019)	A dataset consisting of 3000 sample images of lichen planus, acne and sjs ten	CNN	Accuracy = 96%
3	Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data[25]	Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., ... Bakas, S. (2020)	CDS training data	CDS, FL, CIIL	CDS global Acc. = 86.2% FL global Acc. = 85.7% CLL global Acc. = 85.3%
4	Our work		Medical MNIST	Feature extraction: resNet50, VGG16, Inception-v3 Models: CNN, MLP,FL	CNN + FL= 99.661% CNN = 97.22% MLP + FL =98.592% MLP = 97.83%

Table 4.6: Comparison with related works

# Chapter 5

## Conclusion and Future Work

Federated learning on a hybrid database that has been pre-trained is used to increase security for healthcare data in order to minimize the risk of data leakage. Despite having several encryption algorithms, federated learning is prioritized here because it does not necessitate data transmission. To increase healthcare security, the implementation of our proposed hybrid federated learning model is providing better accuracy of 99.661%. Moreover, using the federated learning approach, which involves dispersing applied models over a number of devices while thwarting data theft. The system will be updated to take patient preferences into account and transmit CNN and MLP models in addition to improving the model performance. The purpose of this research was to employ federated learning to create a solution that would help the e-health care sector while ensuring security.

Due to a shortage of resources, a number of tests and experiments have been put off until the future. If any data is lost throughout the procedure, it can be challenging to retrieve it. We would like to increase model performance relative to models trained on a centralized dataset; however, because of the disparity of the data, it can be difficult to carry out the process. There is still space for improvement on device heterogeneity, as several edge devices operate concurrently in federated learning and their computational power and processing speed are not uniform.

# Bibliography

- [1] H. A. Bourlard and N. Morgan, “Feature extraction by mlp,” in *Connectionist Speech Recognition: A Hybrid Approach*. Boston, MA: Springer US, 1994, pp. 253–263, ISBN: 978-1-4615-3210-1. DOI: 10.1007/978-1-4615-3210-1\_14. [Online]. Available: [https://doi.org/10.1007/978-1-4615-3210-1\\_14](https://doi.org/10.1007/978-1-4615-3210-1_14).
- [2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [3] K. A. Hughes, “# Ilooklikeasurgeon goes viral: How it happened.,” *Bulletin of the American College of Surgeons*, vol. 100, no. 11, pp. 10–16, 2015.
- [4] F. Learning, “Collaborative machine learning without centralized training data,” *Publication date: Thursday, April*, vol. 6, 2017.
- [5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.
- [6] M. Almulhim and N. Zaman, “Proposing secure and lightweight authentication scheme for iot based e-health applications,” in *2018 20th International Conference on advanced communication technology (ICACT)*, IEEE, 2018, pp. 481–487.
- [7] M. Asif-Ur-Rahman, F. Afsana, M. Mahmud, *et al.*, “Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4049–4062, 2018.
- [8] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, “Federated learning of predictive models from federated electronic health records,” *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.
- [9] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, X. Jiang, *et al.*, “Privacy-preserving patient similarity learning in a federated environment: Development and analysis,” *JMIR medical informatics*, vol. 6, no. 2, e7744, 2018.
- [10] S. Bhadula, S. Sharma, P. Juyal, and C. Kulshrestha, “Machine learning algorithms based skin disease detection,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 2, pp. 4044–4049, 2019.
- [11] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, “Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data,” in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, IEEE, 2019, pp. 270–274.

- [12] F. Wang, H. Zhu, X. Liu, *et al.*, “Privacy-preserving collaborative model learning scheme for e-healthcare,” *IEEE Access*, vol. 7, pp. 166 054–166 065, 2019.
- [13] T. Benil and J. Jasper, “Cloud based security on outsourcing using blockchain in e-health systems,” *Computer Networks*, vol. 178, p. 107 344, 2020.
- [14] N. Deepa and P. Pandiaraja, “Hybrid context aware recommendation system for e-health care by merkle hash tree from cloud using evolutionary algorithm,” *Soft Computing*, vol. 24, no. 10, pp. 7149–7161, 2020.
- [15] S. M. E. B. R. GA, “Martin j pati s kotrotsou a milchenko m xu w marcus d colen rr et al,” *Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data Scientific Reports*, vol. 10, no. 1, p. 1, 2020.
- [16] R. Hamza, Z. Yan, K. Muhammad, P. Bellavista, and F. Titouna, “A privacy-preserving cryptosystem for iot e-healthcare,” *Information Sciences*, vol. 527, pp. 493–510, 2020.
- [17] Larxel, *Medical mnist*, Apr. 2020. [Online]. Available: [https://www.kaggle.com/datasets/andrewmvd/medical-mnist?fbclid=IwAR2WsJJvYclY5FTBhVUa6yMsc3xB-WUFRWwzxq9CP\\_k9GLEGQtwu5MZuJZI](https://www.kaggle.com/datasets/andrewmvd/medical-mnist?fbclid=IwAR2WsJJvYclY5FTBhVUa6yMsc3xB-WUFRWwzxq9CP_k9GLEGQtwu5MZuJZI).
- [18] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, “Heart disease identification method using machine learning classification in e-healthcare,” *IEEE Access*, vol. 8, pp. 107 562–107 582, 2020.
- [19] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, “Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results,” *Medical Image Analysis*, vol. 65, p. 101 765, 2020.
- [20] K. G. Ntonja, G. M. Muketha, and G. N. Kamau, “Cloud data privacy preserving model for health information systems based on multi factor authentication,” 2020.
- [21] A. M. Perumal and E. R. S. Nadar, “Architectural framework of a group key management system for enhancing e-healthcare data security,” *Healthcare Technology Letters*, vol. 7, no. 1, pp. 13–17, 2020.
- [22] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, “Secure and robust machine learning for healthcare: A survey,” *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 156–180, 2020.
- [23] N. Rieke, J. Hancox, W. Li, *et al.*, “The future of digital health with federated learning,” *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.
- [24] S. Sengan, G. Kamalam, J. Vellingiri, J. Gopal, P. Velayutham, V. Subramaniaswamy, *et al.*, “Medical information retrieval systems for e-health care records using fuzzy based machine learning model,” *Microprocessors and Microsystems*, p. 103 344, 2020.
- [25] M. J. Sheller, B. Edwards, G. A. Reina, *et al.*, “Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data,” *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.
- [26] H. Xu, J. Li, H. Xiong, and H. Lu, “Fedmax: Enabling a highly-efficient federated learning framework,” in *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, IEEE, 2020, pp. 426–434.

- [27] M. Zhang, Y. Chen, and W. Susilo, “Ppo-cpq: A privacy-preserving optimization of clinical pathway query for e-healthcare systems,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 660–10 672, 2020.
- [28] M. B. Alazzam, H. Al Khatib, W. T. Mohammad, and F. Alassery, “E-health system characteristics, medical performance, and healthcare quality at jordan’s health centers,” *Journal of healthcare engineering*, vol. 2021, 2021.
- [29] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, “Internet of things (iot): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities,” *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 474–10 498, 2021.
- [30] D. Chen, C. S. Hong, Y. Zha, Y. Zhang, X. Liu, and Z. Han, “Fedsvrg based communication efficient scheme for federated learning in mec networks,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7300–7304, 2021.
- [31] I. Dayan, H. R. Roth, A. Zhong, *et al.*, “Federated learning for predicting clinical outcomes in patients with covid-19,” *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.
- [32] L. Jurado Pérez and J. Salvachúa, “An approach to build e-health iot reactive multi-services based on technologies around cloud computing for elderly care in smart city homes,” *Applied Sciences*, vol. 11, no. 11, p. 5172, 2021.
- [33] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, “Federated-learning-based anomaly detection for iot security attacks,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021.
- [34] M. Muthuppalaniappan and K. Stevenson, “Healthcare cyber-attacks and the covid-19 pandemic: An urgent threat to global health,” *International Journal for Quality in Health Care*, vol. 33, no. 1, mzaa117, 2021.
- [35] B. Pfitzner, N. Steckhan, and B. Arnrich, “Federated learning in a medical context: A systematic literature review,” *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–31, 2021.
- [36] C.-R. Shyu, K. T. Putra, H.-C. Chen, *et al.*, “A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications,” *Applied Sciences*, vol. 11, no. 23, p. 11 191, 2021.
- [37] T. Sun, D. Li, and B. Wang, “Decentralized federated averaging,” *arXiv preprint arXiv:2104.11375*, 2021.
- [38] B. Xi, S. Li, J. Li, H. Liu, H. Liu, and H. Zhu, “Batfl: Backdoor detection on federated learning in e-health,” in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, IEEE, 2021, pp. 1–10.
- [39] Z. Xu, Y. Guo, C. Chakraborty, Q. Hua, S. Chen, and K. Yu, “A simple federated learning-based scheme for security enhancement over internet of medical things,” *IEEE Journal of Biomedical and Health Informatics*, 2022.
- [40] T. Zhou, C. Xu, C. Wang, *et al.*, “Burnout and well-being of healthcare workers in the post-pandemic period of covid-19: A perspective from the job demands-resources model,” *BMC Health Services Research*, vol. 22, no. 1, pp. 1–15, 2022.
- [41] [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations>.



- [42] *The importance of securing healthcare data for patients.* [Online]. Available: <https://www.weforum.org/agenda/2022/08/the-importance-of-securing-healthcare-data/>..
- [43] J. Xu and F. Wang, “Federated learning for healthcare informatics. arxiv 2019,” *arXiv preprint arXiv:1911.06270*,