Spring 2023

Federated Learning for Protecting Medical Data Privacy

Abhishek Reddy Punreddy
San Jose State University

Federated Learning for Protecting Medical Data Privacy

A Project

Presented To

Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Abhishek Reddy Punreddy

May 2023

The Designated Project Committee Approves the Project Titled

Federated Learning for Protecting Medical Data Privacy

by

Abhishek Reddy Punreddy

Approved for the Department of Computer Science

San José State University

May 2023

Professor Robert Chun	Department of Computer Science
Professor Nada Attar	Department of Computer Science
Mr. Hemant Koti	Member of Technical Staff, Salesforce

ABSTRACT

Deep learning is one of the most advanced machine learning techniques, and its prominence has increased in recent years. Language processing, predictions in medical research and pattern recognition are few of the numerous fields in which it is widely utilized. Numerous modern medical applications benefit greatly from the implementation of machine learning (ML) models and the disruptive innovations in the entire modern health care system. It is extensively used for constructing accurate and robust statistical models from large volumes of medical data collected from a variety of sources in contemporary healthcare systems [1]. Due to privacy concerns that restrict access to medical data, these Deep learning techniques have yet to completely exploit medical data despite their immense potential benefits. Many data proprietors are unable to benefit from large-scale deep learning due to privacy and confidentiality concerns associated with data sharing. However, without access to sufficient data, Deep Learning will not be able to realize its maximum potential when transitioning from the research phase to clinical practice [2]. This project addresses this problem by implementing Federated Learning and Encrypted Computations on text data, such as Multi Party Computation. SyferText, a Python library for privacy-protected Natural Language Processing that leverages PySyft to conduct Federated Learning, is used in this context.

Index terms **– Deep Learning, Privacy preserving, Machine Learning, Language Processing, Federated Learning, Multi-Part Computation.**

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Robert Chun for the constant support, positivity, and motivation he has provided me throughout the project. His advice has helped me immensely and directed me in the right direction.

I would also like to express my gratitude towards Prof. Nada Attar and Mr. Hemant Koti for graciously agreeing to be my committee members and giving me valuable feedback.

Finally, I'd like to thank my family and friends for all the support throughout my career and without whom none of this would have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

Deep learning has received a great deal of attention in the scientific community because it enables traditional learning algorithms to surmount their reliance on hand-designed features. Deep learning models can now be used in a variety of domains, including big data analytics and applications such as natural language processing, speech recognition, computer vision, pattern recognition, and intrusion detection, among others, due to their unprecedented accuracy. One such field in which these models have been successfully employed is the medical analysis. Any aspect of the medical analysis that involves manual intervention and repetitive work can cause a lot of fatigue as it can be a laborious task. This can cause human error as a single mistake is all it takes to an incorrect diagnosis of the patient which can sometimes potentially lead to human life loss. While humans may not always perform repetitive tasks to the best of their abilities, machines will perform them tirelessly and consistently [3]. This is where Deep Learning models have had enormous success, not only in Medical Image Analysis but also in other areas of medical science. Monitoring chronic diseases [4, 5], cancer prediction [5, 6], and tumor detection [7, 8] are just a few examples of how ML and DL-based models have revolutionized healthcare.

Deep learning models usually require a large set of data and highly efficient machines to perform any kind of operations. This might not be possible at smaller institutions as they cannot necessarily afford the infrastructure whereas the larger medical institutes can employ their DL models and train them locally. Even these models cannot be termed perfect as they could be biased because of the homogeneous pattern of the data and thus it is not a good representation of the general population [7, 8, 9]. Institutes specializing in a certain disease,

gathering data from a particular geographical location, or not having data of their patients across all ages can lead to data homogeneity and these don't totally help the DL models as they need to be trained across varied types of data. Inorder to achieve this, data needs to be collected from several institutes and this is what exactly raises the privacy concerns.

Centralized training requires gathering data from multiple sources to a single server. Once this is done, the concerned parties lose their ownership and the governance rights [10]. Furthermore, there is no guarantee that the data is securely transmitted and stored, making it vulnerable to attacks. Attempts have been made to create centralized repositories containing anonymized medical data [11-27].

However, data privacy and protection laws such as the GDPR in Europe and HIPAA in the United States impose significant costs on the development of such data repositories [28]. The information must be anonymized so that it cannot be traced back to the original patient. While this is a step toward protecting patients' privacy, the anonymization and de-identification process has a negative impact on the utility of future research data. It is widely acknowledged that simply removing identifiable information such as a patient's name or date of birth is not always enough to protect privacy, even if data anonymization is attempted [29]. In fact, anonymized data can still contain statistical signatures that make it vulnerable to reidentification through linkage attacks [30, 31]. The collection and maintenance of high-quality data sets is a resource-intensive process that requires significant investments of time, effort, and resources. Due to the commercial value of such data, it is less likely to be made freely available and is often subject to the control of data collectors.

Privacy-preserving deep learning is an active area of study that has made significant progress over the past decade. Federated learning permits multiple devices or entities to cooperatively train a model while preserving the confidentiality of their raw data. The model is trained on decentralized data sources. Distributed learning addresses the data ownership and governance concerns expressed by centralized training by moving the model to the data. There are a number of distributed learning methodologies, but they require additional data privacy protection techniques in order to learn without violating the privacy of patients.

II. BACKGROUND

A. Group-Based Anonymity

One of the earliest privacy preservation techniques involved using data anonymization to hide sensitive data. Group-based anonymity is a privacy preservation method used to protect sensitive data shared by multiple individuals. It is a type of k-anonymity approach where data is anonymized by grouping together individuals. Groups are constructed in a way that ensures that each group contains at least k individuals with the same set of attributes or characteristics [35]. This makes it difficult to identify any particular individual in the group, as the group members share the same attributes or characteristics. The method can be implemented using various techniques, such as generalization, suppression, or a combination of both. Generalization involves replacing sensitive data with more general data, while suppression involves removing sensitive data [36-37]. The choice of technique depends on the data being anonymized and the privacy requirements of the application.

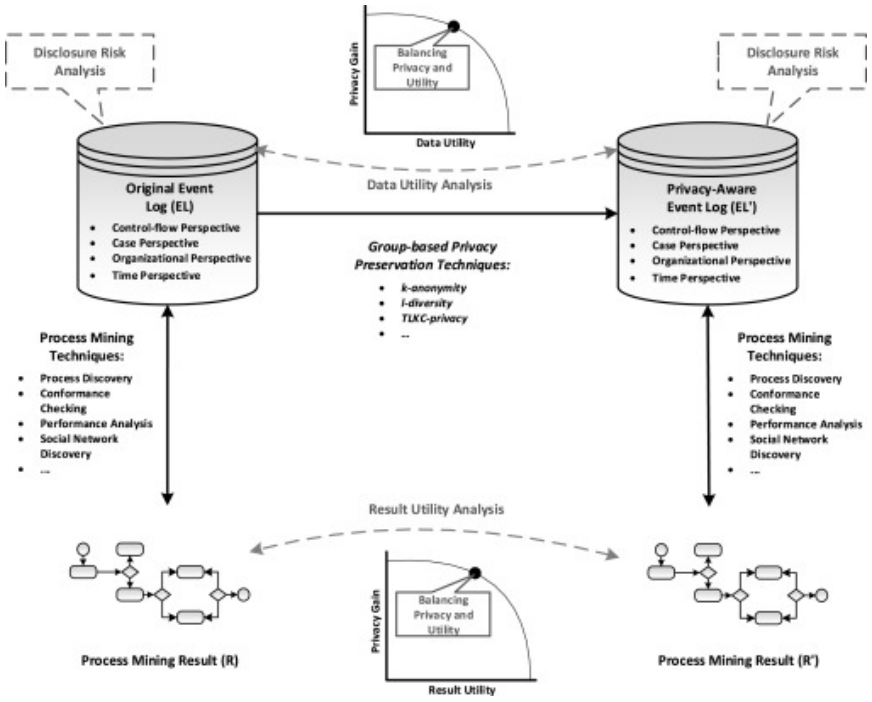

Fig. 1. Multiple Group-based privacy techniques

B. Homomorphic Encryption

Homomorphic encryption is an emerging technique for protecting data privacy while enabling computations to be performed on encrypted data. This method provides a high degree of privacy protection because it encrypts sensitive data throughout the computation process. The encrypted data may be transmitted to a third party for processing without the danger of data exposure or leakage. The result of the computation can only be decrypted by parties with the correct key.

In recent years, the use of homomorphic encryption in real-world applications has grown in popularity. This method has been implemented in numerous industries, including healthcare, finance, and cloud computing. In healthcare, homomorphic encryption has been utilized to protect the privacy of patient medical records, allowing hospitals and healthcare

providers to securely share sensitive health information. Additionally, the technique has the potential to revolutionize finance by facilitating encrypted financial transactions without disclosing sensitive data.

Despite its benefits, homomorphic encryption continues to confront obstacles. The computational burden associated with the encryption and decryption procedures is one of the primary obstacles. These processes can be sluggish and resource-intensive, which can have a significant impact on system efficiency. Researchers are continually enhancing the efficacy of homomorphic encryption techniques to make them more applicable in the real world. In addition, the use of homomorphic encryption in large-scale data processing applications is still in the experimental stage, and more research is required to determine the complete potential of this method.

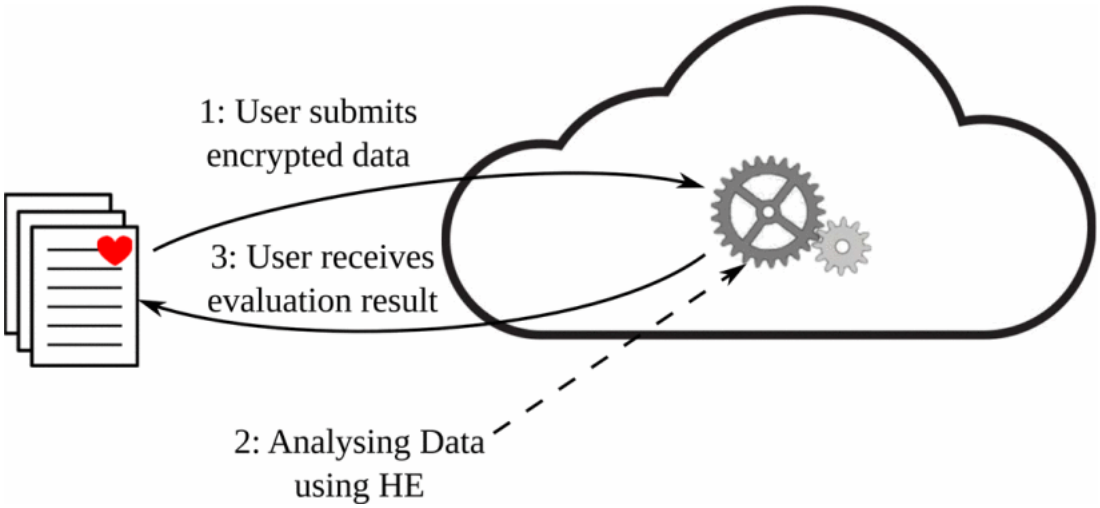

Fig.2. Homomorphic Encryption function on cloud

C. Differential Privacy

Differential privacy is a technique that adds noise to data prior to sharing it in order to safeguard the privacy of the individuals whose data is being shared. The quantity of added noise is meticulously calibrated to not only prevent any individual from getting identified but also to conserve important properties of the data that is in use. Even if an adversary has access to external information, the technique provides a strong mathematical guarantee that the shared data cannot be linked to a specific individual. This procedure is especially useful for protecting sensitive data, such as medical records and financial records.

Differential privacy accomplishes its privacy-protecting objectives by introducing random noise into shared data. The noise is added so that the statistical properties of the data are essentially unaffected, while a high level of privacy is maintained. The amount of added noise is meticulously regulated, and the level of privacy can be altered by adjusting the amount of noise. This method has been proven effective in a variety of real-world situations, including the sharing of medical data for research purposes.

D. Federated Learning

Federated Learning [40] is a machine learning architecture in which numerous devices (such as mobile phones, computers, businesses, etc.) work together to jointly train a learning model under the management of a central server. The primary characteristic of FL is that the dataset is decentralized; each device trains the model on its own local dataset before sending the updated parameters (such as gradient) to a centralized server.

In Fig. 3 [40], a traditional Federated Learning (FL) architecture is displayed. The FL network's clients receive the AI/ML model parameters through broadcast from a single, central

global server. Either a client selection algorithm or a random process could be used by the central server to choose the clients. After receiving the global model's parameters from the global server, the selected clients train the model locally using their own data. After receiving the clients' local models, the server will use their parameters to construct the global model.

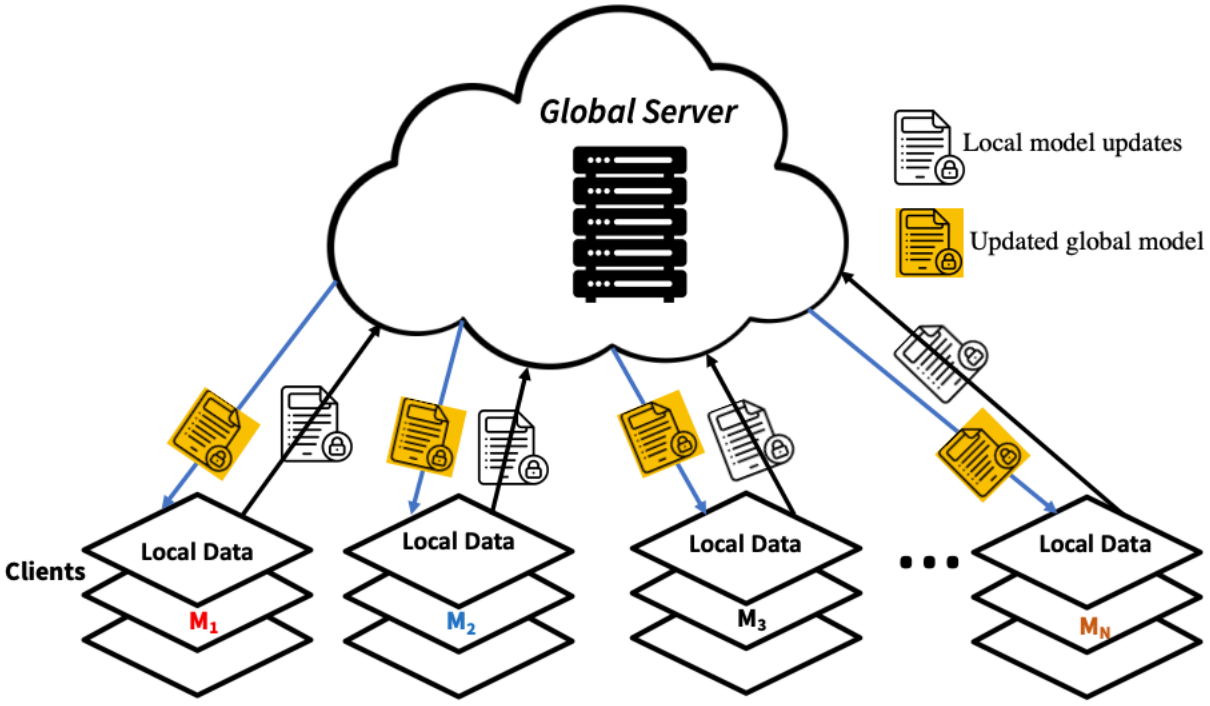

Fig.3. Architecture of FL. The clients transmit to the server local model updates trained with the local dataset for aggregation. Finally, the central server aggregates the local models transmitted by the participating clients and transmits the most recent global model to each participating client.

Fig.4 [40] below illustrates a similar architecture for FL in the medical field, where potential consumers include providers of home healthcare, hospital healthcare, and mobile healthcare. These domains' data distributions could vary. FL can be used in various fields to address issues with security, privacy, healthcare systems, and device distribution by considering how their data is distributed.

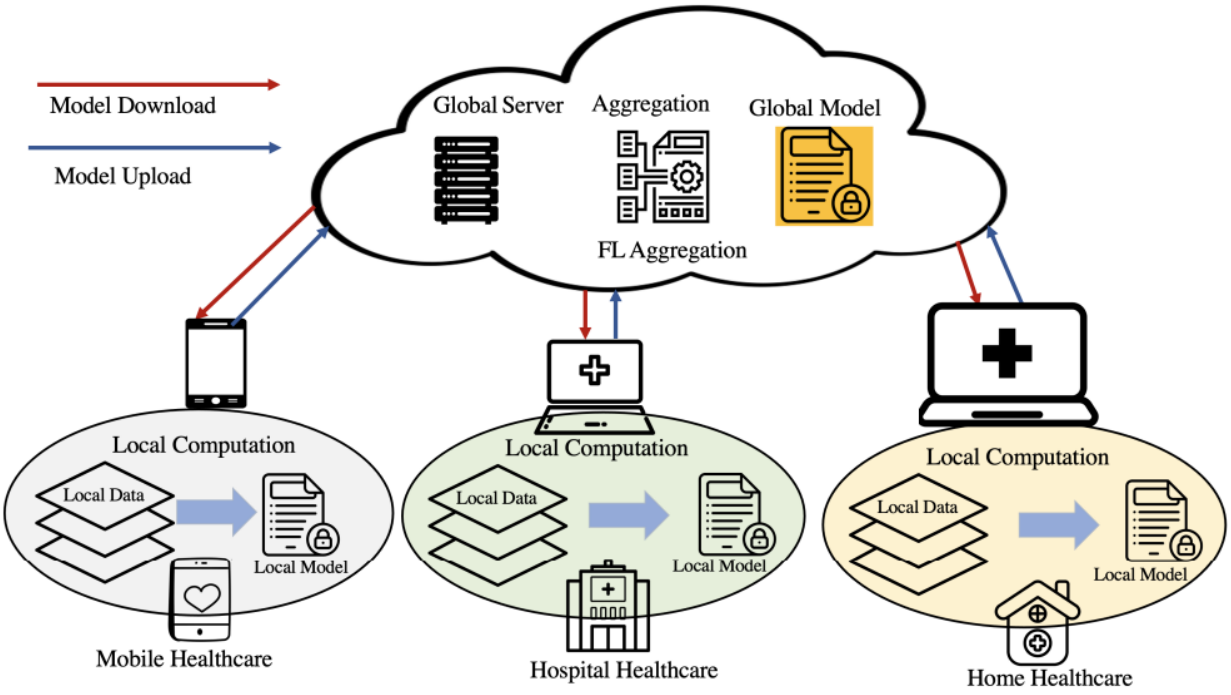

Fig. 4: FL framework for the medical domain.

III. RELATED WORK

Prior to the advent of deep learning, and to this day, anonymization or de-identification of data has remained the most popular method of protecting patients' privacy when sharing medical data. While there are no standardized methods for de-identification and different policies propose different requirements, three main approaches have been identified: 1. removal of patient identifiers or de-identification 2. pseudonymization, or the replacement of patient identifiers with unique pseudonyms, and 3. anonymization, which entails de-identification followed by the removal of additional information to reduce the likelihood of re-identification. Because of its simplicity and the fact that it is built into existing medical image analysis tools, anonymization remains popular.

Anonymization as a means of protecting personal information is explored in [41]. Data randomization is used to sever ties between attribute values in records, hence protecting users' privacy. Homomorphic election models, which allow for several candidates to run in one election, are discussed in [42], along with the necessary components for producing random forests classification with enhanced prediction performance. Different classifier-based approaches to privacy protection have also been suggested. Randomized Response with Partial hiding (RRPH) is a new randomization method introduced in [43] that combines data modification with data hiding to affect the original data. The RRPH-generated distorted data is then used to train a Naive Bayes classifier that accurately predicts the class labels of unknown samples. Best randomization algorithms for privacy-protective density estimation are proposed in [44].

The use of BiBoost and MultBoost algorithms allows boosting classifiers to be constructed without the need for explicit data sharing between multiple participants, as explained in [45], which also examines the computational and security implications of these methods. However, other solutions have been proposed for private computation without revealing sensitive data. For instance, [46] proposes secure sum computation, while [47] adds noise to the source data and [48] utilizes cryptographic tools to construct a decision tree classifier in a secure and efficient manner. In [49], privacy preserving SVM is established as a viable solution, although its protocols rely on circuit evaluation, which is too expensive for most practical applications. In contrast, [50] provides a protocol for the secure computation of any polynomial function in the probabilistic setting. Although in theory, safe multiparty computation solves all privacy-preserving computation problems, in practice it is often unfeasible due to its high cost.

Meanwhile, [51] suggests a cryptographic protocol for the classification of nonlinear data utilizing feedforward neural networks. The three algorithms presented in [52] aim to safeguard private weight vectors and activation functions and prevent data providers from introducing false data into the system. Nevertheless, exchanging data across databases remains a challenge, as noted by [53]. This method is also limited to two parties and lacks a trusted third party, which can pose significant security risks if the querying party is malicious. Furthermore, if the query result is computed by combining data from multiple sources on the SSN field, the querying party will be able to identify the patients using the query result. [54] recommends a similar encryption-based approach for disseminating k-anonymous, de-identified healthcare

data that cannot be linked to publicly available patient identification data. Nonetheless, this method fails to resolve the issue of data integration.

In [55], RG-RP scheme, a privacy-preserving method for protecting against maximum a posteriori (MAP) estimate assaults using recurrent Gompertz (RG) nonlinear perturbation is offered. In contrast, in [56] two polynomial approximation-coupled systems are proposed, with cloud computing carrying out the learning process from encrypted datasets sent by participants over the Secure Multiparty Computation (SMC) protocol.

The concept of "partial parameter sharing" is used in [57] to facilitate collaborative model learning, with each participant training a subset of parameters and then passing along the resulting gradients. After each round of local training in [58], participants submit encrypted local gradients to the cloud. Homomorphic ciphertexts are kept secure by the usage of individual TLS/SSL encrypted channels. The technique proposed in [59] involves users training locally on their own private datasets before sending their perturbed and encrypted local gradients to the cloud. After decryption, users can make changes to their models based on the global gradients.

An alternate technique is proposed in [60], where the trainers instead of sharing the gradients, exchange the model weights, which they claim is more secure against information leakage. Last but not least, [61] employs an approach based on Secure Multiparty Computing (SMC), in which users evaluate their local models, modifications are aggregated securely by a third party, and the combined model is then uploaded to the server. To hide inputs, this technique uses masking with a single time pad generated by adding and subtracting random masking vectors.

IV. PROPOSED METHOD

This paper investigates methods of implementing secure and private Deep Learning by utilizing PySyft and SyferText libraries, which enables separation of sensitive data from the model training process. We present a classifier that can accurately categorize medical specialties based on transcription text without direct access to the dataset. Following tasks are performed on the dataset that is chosen.

- Utilizing the PySyft library to combine the client's individual datasets into a larger one.
- Employing the SyferText library to process and ready the text data on the client's machines while keeping it confidential and without transferring any datasets to your machine.

In-order to comprehend the dataset and the operations performed on it, it is crucial to have a thorough understanding of the internal workings of the libraries and the underlying concepts they employ. This section consists of the various concepts that are employed.

4.1 Federated Learning

Federated learning is a decentralized machine learning method that enables multiple devices to collaboratively train a model without sharing their data. The local data of each device is used to train the model, with the resulting updates sent to a central server. These updates are then combined by the server and sent back to the devices for further training, continuing until convergence of the model.

Federated learning can be used to preserve privacy in medical data processing by keeping the patient data on individual devices. The model can be trained on the local data of

each device without the need to transfer the data to a central server. This ensures that the patient data is not shared with third parties, and the privacy of patients is preserved.

4.2 PySyft

PySyft is an open-source library that enables secure and privacy-preserving machine learning (ML) computations using various techniques. The total internal working of PySyft can be described as follows:

- PySyft extends the PyTorch framework: PySyft extends the PyTorch framework, a popular ML library, to enable secure and privacy-preserving computations. PySyft allows PyTorch tensors to be shared between different parties, enabling secure computations across multiple devices.
- PySyft uses Federated Learning: PySyft uses Federated Learning (FL) to train ML models without the need to centralize data. FL allows the model to be trained collaboratively across multiple devices, without sharing the underlying data. PySyft provides a range of tools and techniques to ensure the security and privacy of FL.
- PySyft uses Secure Multi-Party Computation: PySyft uses Secure Multi-Party Computation (MPC) to enable secure computations across multiple parties without the need to share their private data. PySyft supports different MPC protocols, including secret sharing and garbled circuits, to enable secure computation.
- PySyft uses Differential Privacy: To protect sensitive information during training, PySyft employs Differential Privacy (DP). To prevent the model from being properly trained on identifiable data points, DP injects noise into the data.

- PySyft supports Encrypted Computation: PySyft supports Encrypted Computation, allowing for computations to be performed on encrypted data without revealing the data to any party involved in the computation. PySyft supports different encryption techniques, including Homomorphic Encryption, Secure Multiparty Computation, and Private Set Intersection, to enable secure computations on encrypted data.

Overall, PySyft is a powerful and flexible tool for building secure, privacy-preserving ML applications, and is used by researchers and practitioners around the world. Its total internal working involves extending PyTorch, using FL, MPC, DP, and Encrypted Computation to enable secure and privacy-preserving computations.

4.3 SyferText

SyferText is an open-source natural language processing (NLP) library built on top of PySyft, the secure and privacy-preserving deep learning library. SyferText extends the popular spaCy NLP library, providing a similar API and adding privacy-preserving features. SyferText includes various pre-processing techniques, such as tokenization, lemmatization, and part-of-speech tagging.

The NLP (Natural Language Processing) pipeline is a series of sequential steps or processes that are applied to text or speech data to extract meaningful information and insights. The pipeline is a high-level representation of the NLP workflow, which involves a combination of linguistic and machine learning techniques to analyze, process, and understand human language.

4.3.1 Tokenization

Tokenization is the process of breaking down a text document into smaller units called tokens. In Natural Language Processing (NLP), tokenization is often the first step in the NLP pipeline, which is a series of sequential steps or processes that are applied to text or speech data to extract meaningful information and insights. The most common approach to tokenization is word tokenization, which involves splitting a text document into individual words. This is typically done by identifying whitespace and punctuation marks such as commas, periods, and colons as word boundaries.

Tokenization is an essential preprocessing step in many NLP tasks, such as language modeling, part-of-speech tagging, named entity recognition, sentiment analysis, and machine translation. It enables computers to process and understand human language by breaking it down into smaller, more manageable units that can be analyzed and processed using a variety of NLP techniques.

4.3.2 Part-of-Speech Taggers

Part-of-speech (POS) tagging is the process of assigning a grammatical tag to each word in a sentence, indicating its syntactic role in the sentence and its relationship with other words. POS tagging involves analyzing the structure of a sentence and identifying the word's function in that sentence. Each word in a sentence is assigned a specific tag that indicates its part of speech, such as noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. These tags provide valuable information about the syntactic structure of the sentence and are useful in a variety of natural language processing applications, such as language translation, information retrieval, and speech recognition.

Vocab taggers and stop taggers are both specialized types of POS taggers that focus on specific aspects of POS tagging. A vocab tagger uses a pre-defined list of words and their corresponding POS tags to assign tags to new words based on their similarity to the words in the list. This is especially useful for handling out-of-vocabulary (OOV) words that are not present in a pre-trained model. Instead of relying on complex statistical models to assign tags to OOV words, a vocab tagger can simply look up the word in the lexicon and assign the corresponding POS tag. On the other hand, stop words are common words that are often removed from a text corpus because they do not provide significant information about the content of the text. Examples of stop words include "the", "and", "a", "of", "in", "to", "is", etc. A stop tagger is designed to identify and tag these stop words with a specific POS tag, typically a "STOP" tag or an empty tag, to indicate that they should be removed from the text corpus.

V. EXISTING TOOLS

While PySyft is unique in many ways, there are several existing tools in the industry that share similar goals and features such as TensorFlow Privacy, IBM Differential Privacy Library, Microsoft SEAL etc. This section reviews these existing tools and their internal working and compares them to PySyft.

5.1 TensorFlow Privacy

TensorFlow Privacy is an open-source library that provides tools and techniques for training machine learning models with differential privacy. TensorFlow Privacy provides several tools and techniques for achieving differential privacy during model training. These include Gaussian noise, Sampled Gaussian mechanism, and optimization algorithms like stochastic gradient descent (SGD) and Adam optimizer. While TensorFlow Privacy is a powerful and flexible tool for building privacy-preserving machine learning models, there are some drawbacks when compared to PySyft as follows

- It is built on top of the TensorFlow framework, which can make it more complex and harder to use than PySyft.
- While it provides powerful tools for adding differential privacy to TensorFlow models, it is more limited in scope than PySyft
- It is designed to work with the TensorFlow framework, which can limit its compatibility with other machine learning frameworks and libraries. In contrast, PySyft is designed to be more flexible and can be used with a range of machine learning frameworks and libraries, including TensorFlow.

5.2 IBM Differential Privacy

IBM Differential Privacy Library is an open-source library that provides tools and techniques for implementing differential privacy in machine learning applications. The library is designed to be easy to use, flexible, and efficient, enabling developers to add privacy protection to their models without sacrificing accuracy or performance. It provides several techniques for achieving privacy in machine learning applications including Laplace Mechanism, Exponential Mechanism and Restricted Sensitivity Mechanism. While it is a powerful tool for adding differential privacy to machine learning models, there are some drawbacks when compared to PySyft as follows

- It is limited in scope as it is designed specifically for adding differential privacy to machine learning models, while PySyft provides a range of privacy-preserving techniques, including homomorphic encryption and multi-party computation, as well as support for a range of communication protocols.
- It can be more complex to use than PySyft due to its focus on differential privacy. PySyft provides a simpler, high-level API that abstracts away much of the underlying complexity.
- It is designed to work with the IBM Watson Machine Learning service, which can limit its compatibility with other machine learning frameworks and libraries. In contrast, PySyft is designed to be more flexible and can be used with a range of machine learning frameworks and libraries.

5.3 Microsoft SEAL

Microsoft SEAL (Simple Encrypted Arithmetic Library) is an open-source Homomorphic Encryption (HE) library developed by Microsoft Research. Computations can be conducted on encrypted data using HE, eliminating the need for data decryption. SEAL is designed to be efficient, fast and flexible, allowing computations on encrypted data to be performed at scale. Three main components of it are Encryption, Evaluation and Decryption. Encryption process involves generating keys, encoding and encryption. Evaluation process involves Homomorphic operations & noise management. Decryption process involves decryption and decoding. This library too has some drawbacks when compared to PySyft as follows:

- It is designed specifically for homomorphic encryption, while PySyft provides a range of privacy-preserving techniques.
- Homomorphic encryption can be a complex and computationally expensive technique, and Microsoft SEAL is no exception.
- SEAL is designed to work with C++ and .NET programming languages, which can limit its compatibility with other machine learning frameworks and libraries.

VI. DATASET

Data from 40 medical specialties, comprising around 5000 transcribed medical reports, was obtained from https://www.mtsamples.com. The original data was pre-processed and converted into a CSV format for research. These reports serve as samples for reference purposes and are provided by various transcriptionists and users. It is difficult to obtain medical data in real-world situations due to privacy regulations.

6.1 Exploratory Data Analysis

Data stored in csv file has different features of which two have been identified as very essential for training.

- transcription: The medical transcription text
- medical_specialty: The medical specialty tagged to the corpus

There are 40 different medical specialties with different frequencies as can be seen in the picture below

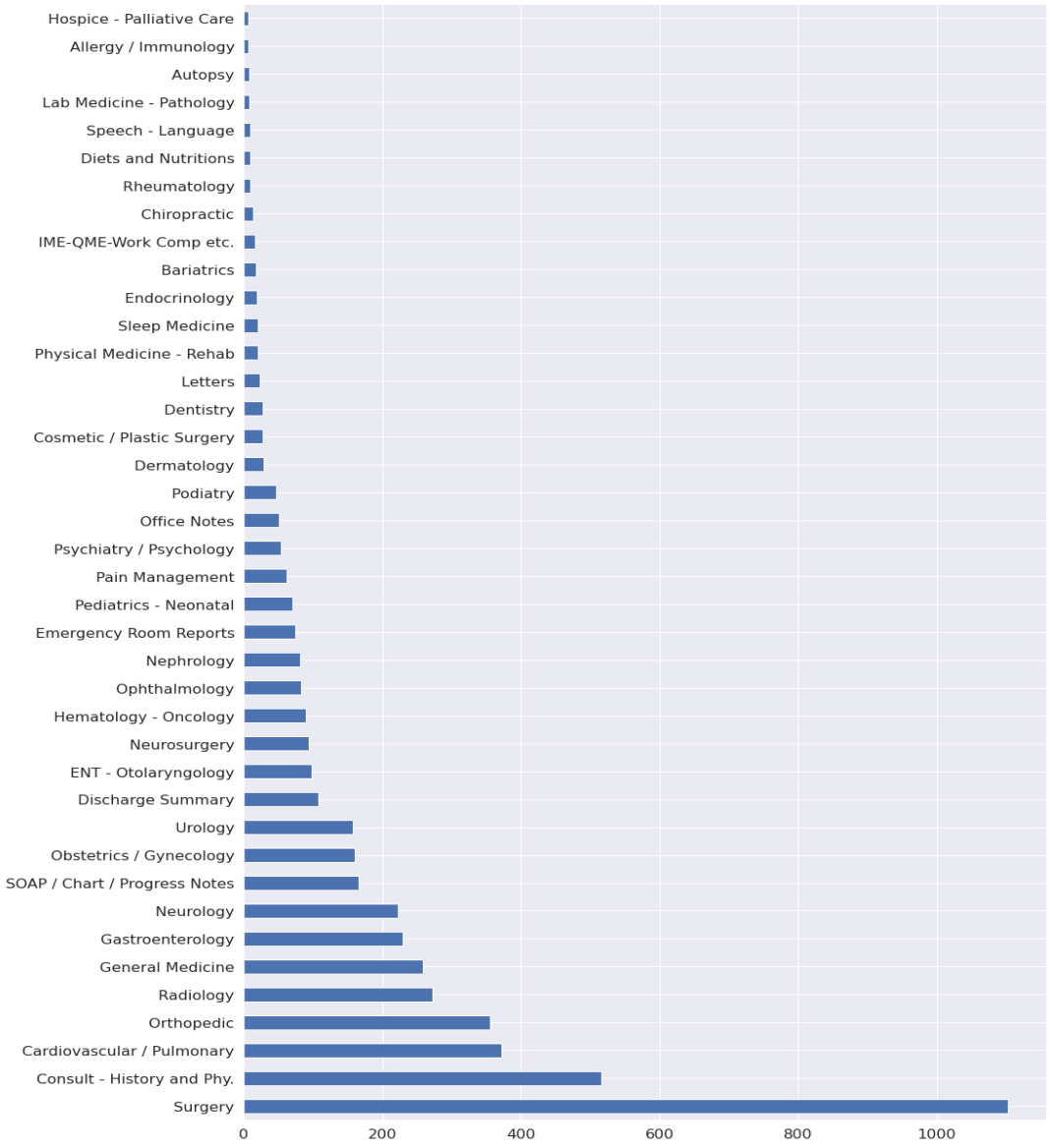

Fig 5. List of medical specialties and their frequencies

Density of the medical_specialty is not evenly distributed as few of the specialties have very high frequency and many others with very less frequency thus making the data skewed. When the data is skewed in machine learning, it can have a significant impact on the performance of the model. Skewed data is characterized by an imbalanced distribution of the target variable. This means that one class is significantly overrepresented, while the other class is significantly underrepresented.

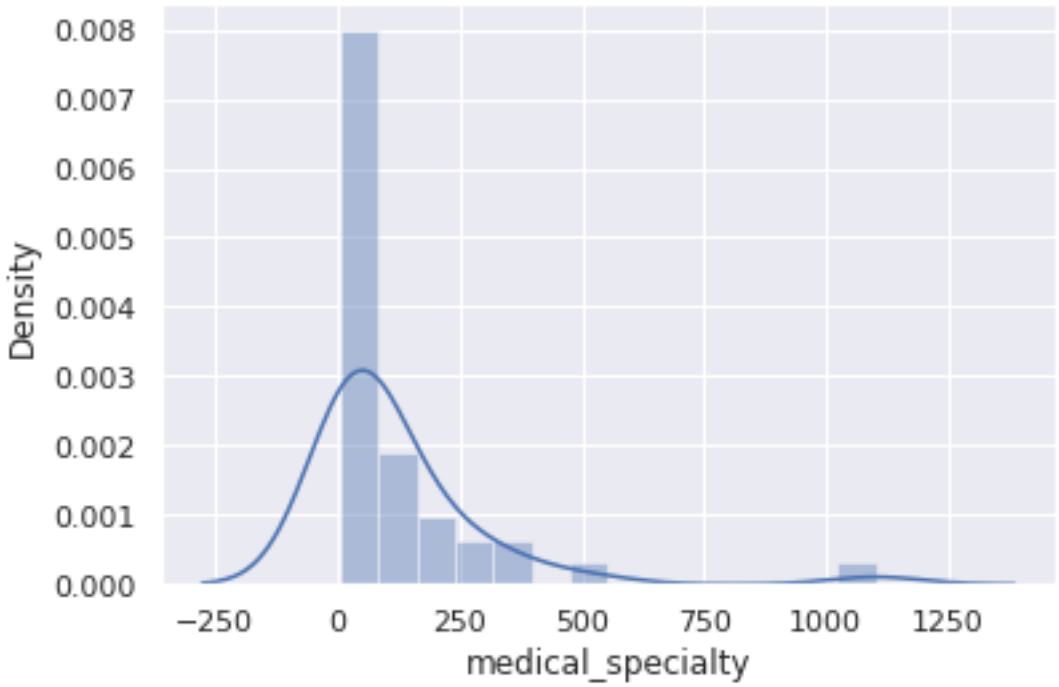

Fig 6. Density distribution of the medical specialties

```
def normality_testing(sk, kt):
    sk = round(sk, 3)
    kt = round(kt, 3)
    if kt > 0:
        print("positive kurtosis =",kt)
    else:
        print("negative kurtosis (less normal) = ",kt)

    if abs(sk) > 1:
        print("skewness =",sk,"\nThe distribution is highly skewed")
    elif abs(sk >= 0.5):
        print("skewness =",sk,"\nThe distribution is moderately skewed")
    else:
        print("skewness =",sk,"\nThe distribution is approximately symmetric")
```

```
skewness = df['medical_specialty'].value_counts().skew()
kurtosis = df['medical_specialty'].value_counts().kurt()

normality_testing(skewness, kurtosis)
```

```
positive kurtosis = 15.336
skewness = 3.516
The distribution is highly skewed
```

Fig 7. Data distribution skewness

6.2 Dimensionality Reduction

The initial dataset has a lot of classes, but the surgery class has a much higher frequency. To address this, we reduced the number of features to distinguish between surgery and non-surgery transcription texts. All the specialties have been condensed into 4 classes in the following manner.

```
new_class = 'Internal Medicine'

df[feature].mask(df[feature] == 'Hospice - Palliative Care', new_class, inplace=True)
df[feature].mask(df[feature] == 'Pain Management', new_class, inplace=True)
df[feature].mask(df[feature] == 'Sleep Medicine', new_class, inplace=True)
df[feature].mask(df[feature] == 'Endocrinology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Gastroenterology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Hematology - Oncology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Nephrology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Rheumatology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Cardiovascular / Pulmonary', new_class, inplace=True)
df[feature].mask(df[feature] == 'General Medicine', new_class, inplace=True)
```

Fig 8. Classification of Internal Medicine

```
new_class = 'Surgery'

df[feature].mask(df[feature] == 'Surgery', new_class, inplace=True)
df[feature].mask(df[feature] == 'Cosmetic / Plastic Surgery', new_class, inplace=True)
df[feature].mask(df[feature] == 'Neurosurgery', new_class, inplace=True)
df[feature].mask(df[feature] == 'ENT - Otolaryngology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Obstetrics / Gynecology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Urology', new_class, inplace=True)
```

Fig 9. Classification of Surgery

```
new_class = 'Medical Records'

df[feature].mask(df[feature] == 'Consult - History and Phy.', new_class, inplace=True)
df[feature].mask(df[feature] == 'Discharge Summary', new_class, inplace=True)
df[feature].mask(df[feature] == 'Emergency Room Reports', new_class, inplace=True)
df[feature].mask(df[feature] == 'IME-QME-Work Comp etc.', new_class, inplace=True)
df[feature].mask(df[feature] == 'Letters', new_class, inplace=True)
df[feature].mask(df[feature] == 'Office Notes', new_class, inplace=True)
df[feature].mask(df[feature] == 'SOAP / Chart / Progress Notes', new_class, inplace=True)
df[feature].mask(df[feature] == 'Radiology', new_class, inplace=True)
```

Fig 10. Classification of Medical Records

```
new_class = 'Other'

df[feature].mask(df[feature] == 'Diets and Nutritions', new_class, inplace=True)
df[feature].mask(df[feature] == 'Bariatrics', new_class, inplace=True)
df[feature].mask(df[feature] == 'Dentistry', new_class, inplace=True)
df[feature].mask(df[feature] == 'Ophthalmology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Pediatrics - Neonatal', new_class, inplace=True)
df[feature].mask(df[feature] == 'Dermatology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Allergy / Immunology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Speech - Language', new_class, inplace=True)
df[feature].mask(df[feature] == 'Psychiatry / Psychology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Autopsy', new_class, inplace=True)
df[feature].mask(df[feature] == 'Lab Medicine - Pathology', new_class, inplace=True)
df[feature].mask(df[feature] == 'Physical Medicine - Rehab', new_class, inplace=True)
df[feature].mask(df[feature] == 'Orthopedic', new_class, inplace=True)
df[feature].mask(df[feature] == 'Chiropractic', new_class, inplace=True)
df[feature].mask(df[feature] == 'Podiatry', new_class, inplace=True)
df[feature].mask(df[feature] == 'Neurology', new_class, inplace=True)
```

Fig 11. Classification of the rest

We trained a multi-class classifier using the four specialties with the highest frequency as shown in the figure.

Fig 12. Top four specialties

To use SyferText's NLP pipeline, we require the stop words and vocabulary files. For our project, we utilized the clinical concepts repository files, intended for large datasets, and created the vocabulary words from the classes in Systematized Nomenclature of Medicine (SNMI) data.

VII. EXPERIMENTS

Our project involves creating a simulated environment where individual clients possess a portion of the complete dataset. We equip each worker to perform encrypted training on these datasets.

7.1 PySyft

PySyft is used to simulate a hospital environment represented by virtual workers, in which each location stores its datasets locally, without requiring data sharing with a central server. For our specific application, we establish a virtual environment (represented by virtual workers Alice and Bob) using PySyft, which allows us to train our classifier in a secure manner. To create a simulated work environment, three main actors are involved - a company and two clients who possess two separate private datasets named Bob and Alice. In addition, there is a crypto provider who will supply the necessary components for Secure Multi-Party Computation (SMPC).

```
# Create a torch hook for PySyft
hook = sy.TorchHook(torch)

# Create some PySyft workers
me = hook.local_worker # This is the worker representing the deep learning company
bob = sy.VirtualWorker(hook, id = 'bob') # Bob owns the first dataset
alice = sy.VirtualWorker(hook, id = 'alice') # Alice owns the second dataset

crypto_provider = sy.VirtualWorker(hook, id = 'crypto_provider') # provides encryption primitive for SMPC
```

Fig 13. Creation of virtual workers

Using the SyferText library's send() function, we distribute the individual datasets privately to the respective clients. Dataset is split into two parts, one for Bob and the other for Alice.

Split the dataset into two parts, one for Bob and the other for Alice. Each part will be also split into a training set and a validation set. This will create four lists: train_bob, valid_bob, train_alice, valid_alice. Each list has the same format. The dataset has already been divided into train, validation data sets which are assigned to both the workers. Test data has also been separated from the original dataset.

7.2 SyferText NLP Pipeline

The NLP pipeline of SyferText is comprised of three components - a tokenizer, a stop words tagger, and a vocabulary tagger. Additionally, to access all the natural language processing objects and functions, a language object provided by SyferText must be loaded. The NLP object created by SyferText eliminates the stop word tokens added to the pipeline, and only tokens marked as words from the vocabulary file are retained. This pipeline improves text processing efficiency and assigns weights to tokens with a strong correlation to the output classes.

7.3 Encrypted Deep Learning

Our approach involves developing a hook for PyTorch that can be linked to PySyft to broaden the capabilities of PyTorch and make it compatible with PySyft methods. Our network structure is defined, and the data is loaded. By applying PySyft's share() function, the network is shared among the virtual workers. The tensors can be sent to virtual workers using the send(worker) method. The remote operations can be performed on these tensors, in this case, we use forward and backward passes to train our model. Once the operation is completed, we can securely retrieve the tensor by calling the get() function.

7.4 Encrypted Classifier and Hyperparameters

The hyper-parameters used for training and validation are as follows.

- Embedding Dimension: The dimension of the embedding vector for the training dataset.
- Batch Size: 128
- Learning Rate: 0.001
- Output classes: 4

A Linear classifier is created with the following configuration

Layer	In features	Out features	Bias
fc1	300	128	True
fc2	128	64	True
fc3	64	32	True
fc4	32	16	True
Fc5	16	2	True

Table 1: Classifier configuration

The network serves as a classifier with multiple classes, which means it can generate one of four labels - 'Surgery', 'Medical Records', 'Internal Medicine', or 'Other' - based on the transcribed text.

7.5 Results

Once the classifier is prepared, an optimizer is used as well. SGD optimizer is chosen here as shown below

```
optimizer = optim.SGD(params = model.parameters(),lr = LEARNING_RATE, momentum=0.3)
optimizer = optimizer.fix_precision()

print(optimizer)
```

```
SGD (
Parameter Group 0
    dampening: 0
    lr: FixedPrecisionTensor>tensor(1)
    momentum: FixedPrecisionTensor>tensor(300)
    nesterov: False
    weight_decay: 0
)
```

Fig 14. Optimizer configuration

Several attempts at running the model have been done before the model finally obtained an accuracy of approximately 82% as the loss decreased. This could be attributed to our use of the SGD and MSE optimizer, as there are no better alternatives for this framework yet.

Accuracy	Loss
25	110.43
65.62	58.21
69.53	53.36
72.66	42.38
82.03	27.5

Table 2: Accuracy and Loss results

Figure 15. Accuracy graph on training and validation dataset with private classifier

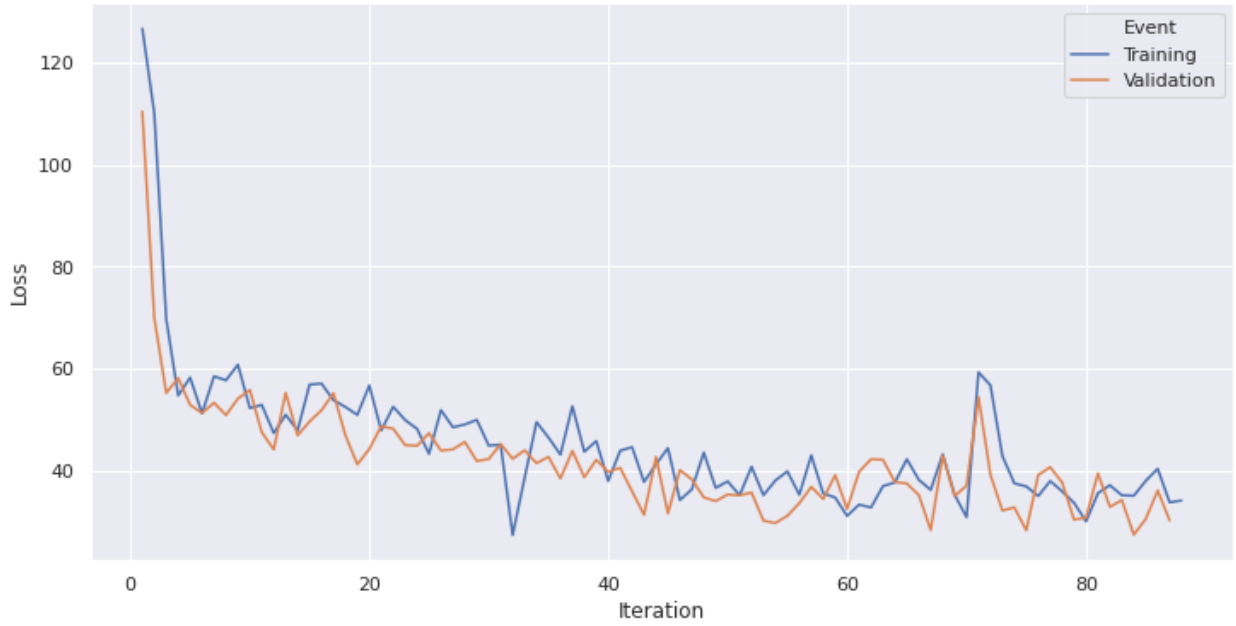

Figure 16. Loss graph on training and validation dataset with private classifier

VIII. CONCLUSION

Deep Neural Networks, for example, have had tremendous success in the medical field, helping to relieve the burden from their human counterparts. However, neural networks are "data hungry," requiring massive amounts of sensitive medical data to learn. Aside from the growing privacy concerns raised by training DL models on private medical data, obtaining such a large amount from a single institute is difficult.

Federated Learning has proven to be a promising solution for preserving privacy in data. It enables multiple parties to train machine learning models collaboratively, without exchanging or centralizing datasets. This approach has the potential to revolutionize various industries by allowing organizations to leverage the collective knowledge of their data without compromising individual privacy. The use of federated learning can also lead to improved accuracy and reduced costs, making it an attractive option for businesses and researchers alike.

While FL is still in its early stages, it has already shown significant potential in the field of privacy-preserving machine learning. As the technology continues to mature and more research is conducted, we can expect to see even greater advancements and applications of it. However, there are still technical challenges that need to be addressed, such as the optimization of communication costs and the development of robust security mechanisms. Overall, the use of Federated Learning as a privacy-preserving technique has great potential and warrants further exploration in order to fully realize its benefits.

IX. FUTURE WORK

In this research, PySyft library has been used which predominantly uses Federated Averaging (FedAvg) as the underlying federated learning algorithm. This is the most popular federated learning technique, in which clients do local updates on their own data and communicate model modifications to the server, which aggregates them using simple averaging. But other algorithms such as Federated Stochastic Gradient Descent (FedSGD) and Federated Averaging with Local Adaption (FedAvgLA) can also be tried to explore the impact on the model's performance and privacy preservation.

The hyper-parameters used in classifiers can be altered as well to see how the model gets affected. The applicability of the current approach can be explored on different kinds of datasets to see if the performance and privacy preservation holds up. There may be opportunities to improve the architecture of the federated learning model utilized in this research. To improve the model's performance and convergence rate, use of various neural network architectures, regularization techniques, or weight initialization strategies could be investigated.

While federated learning is intended to protect privacy, there are possible flaws that malevolent actors could exploit. Future research can be focused on the resilience of privacy preservation in federated learning under various attack scenarios, as well as techniques for minimizing these weaknesses.

REFERENCES

[1] D. Mahendran, C. Luo, and B. T. Mcinnes, “Review: Privacy-Preservation in the Context of Natural Language Processing,” vol. 9. pp. 147600–147612, 2021.

[2] Erickson BJ, Korfiatis P, Akkus Z, Kline TL. Machine Learning for Medical Imaging. Radiographics. 2017 Mar-Apr;37(2):505-515. doi: 10.1148/rg.2017160130. Epub 2017 Feb 17. PMID: 28212054; PMCID: PMC5375621.

[3] O. Aouedi, M. A. Bach Tobji, and A. Abraham, “An Ensemble of Deep Auto-Encoders for Healthcare Monitoring,” presented at the Hybrid Intelligent, 2020, pp. 96–105.

[4] J. Peng, N. Babaguchi, H. Luo, Y. Gao, and J. Fan, “Constructing Distributed Hippocratic Video Databases for Privacy-Preserving Online Patient Training and Counseling,” vol. 14, no. 4. pp. 1014–1026, 2010.

[5] Xiao Y, Wu J, Lin Z, Zhao X. A semi-supervised deep learning method based on stacked sparse auto-encoder for cancer prediction using RNA-seq data. Comput Methods Programs Biomed. 2018 Nov;166:99-105. doi: 10.1016/j.cmpb.2018.10.004. Epub 2018 Oct 5. PMID: 30415723.

[6] L. Zhen and A. Chan, “An artificial intelligent algorithm for tumor detection in screening mammogram,” vol. 20, pp. 559–567, 2001, doi: 10.1109/42.932741.

[7] K. Chang *et al.*, “Distributed deep learning networks among institutions for medical imaging,” vol. 25, no. 8, pp. 945–954, 2018, doi: 10.1093/jamia/ocy017. [Online]. Available: https://doi.org/10.1093/jamia/ocy017

[8] J. Jeon, J. Kim, J. Kim, K. Kim, A. Mohaisen, and J.-K. Kim, “Privacy-Preserving Deep Learning Computation for Geo-Distributed Medical Big-Data Platforms,” presented at the - 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S), 2019, pp. 3–4, doi: 10.1109/DSN-S.2019.00007.

[9] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15). Association for Computing Machinery, New York, NY, USA, 1310–1321. https://doi.org/10.1145/2810103.2813687

[10] P. Mohassel and Y. Zhang, "SecureML: A System for Scalable Privacy-Preserving Machine Learning," 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 19-38, doi: 10.1109/SP.2017.12.

[11] “Ibm’s merge healthcare acquisition.” [Online]. Available: https://www.reuters.com/article/us-merge-healthcare-m-a-ibm/ibm-to-buy-merge-healthcare-in-1-billion-deal-idUSKCN0QB1ML20150806

[12] “Nhs scotland’s national safe haven.” [Online]. Available: https://www.gov.scot/publications/charter-safe-havens-scotland-handling-unconsented-data-national-health-service-patient-records-support-research-statistics/pages/4/

[13] Cuggia M, Combes S. The French Health Data Hub and the German Medical Informatics Initiatives: Two National Projects to Promote Data Sharing in Healthcare. Yearb Med Inform. 2019 Aug;28(1):195-202. doi: 10.1055/s-0039-1677917. Epub 2019 Aug 16. PMID: 31419832; PMCID: PMC6697511.

[14] “Health Data Research UK.” [Online]. Available: https://www.hdruk.ac.uk/

[15] Sporns O, Tononi G, Kötter R. The human connectome: A structural description of the human brain. PLoS Comput Biol. 2005 Sep;1(4):e42. doi: 10.1371/journal.pcbi.0010042. PMID: 16201007; PMCID: PMC1239902.

[16] Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, Downey P, Elliott P, Green J, Landray M, Liu B, Matthews P, Ong G, Pell J, Silman A, Young A, Sprosen T, Peakman T, Collins R. UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Med. 2015 Mar 31;12(3):e1001779. doi: 10.1371/journal.pmed.1001779. PMID: 25826379; PMCID: PMC4380465.

[17] Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. J Digit Imaging. 2013 Dec;26(6):1045-57. doi: 10.1007/s10278-013-9622-7. PMID: 23884657; PMCID: PMC3824915.

[18] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2097–2106.

[19] Yan K, Wang X, Lu L, Summers RM. DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. J Med Imaging (Bellingham). 2018 Jul;5(3):036501. doi: 10.1117/1.JMI.5.3.036501. Epub 2018 Jul 20. PMID: 30035154; PMCID: PMC6052252.

[20] Tomczak K, Czerwińska P, Wiznerowicz M. The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. Contemp Oncol (Pozn). 2015;19(1A):A68-77. doi: 10.5114/wo.2014.47136. PMID: 25691825; PMCID: PMC4322527.

[21] Jack CR Jr, Bernstein MA, Fox NC, Thompson P, Alexander G, Harvey D, Borowski B, Britson PJ, L Whitwell J, Ward C, Dale AM, Felmlee JP, Gunter JL, Hill DL, Killiany R, Schuff N, Fox-Bosetti S, Lin C, Studholme C, DeCarli CS, Krueger G, Ward HA, Metzger GJ, Scott KT, Mallozzi R, Blezek D, Levy J, Debbins JP, Fleisher AS, Albert M, Green R, Bartzokis G, Glover G, Mugler J, Weiner MW. The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods. J Magn Reson Imaging. 2008 Apr;27(4):685-91. doi: 10.1002/jmri.21049. PMID: 18302232; PMCID: PMC2544629.

[22] D. Mahendran, C. Luo and B. T. Mcinnes, "Review: Privacy-Preservation in the Context of Natural Language Processing," in IEEE Access, vol. 9, pp. 147600-147612, 2021, doi: 10.1109/ACCESS.2021.3124163.

[23] Litjens G, Bandi P, Ehteshami Bejnordi B, Geessink O, Balkenhol M, Bult P, Halilovic A, Hermsen M, van de Loo R, Vogels R, Manson QF, Stathonikos N, Baidoshvili A, van Diest P, Wauters C, van Dijk M, van der Laak J. 1399 H&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset. Gigascience. 2018 Jun 1;7(6):giy065. doi: 10.1093/gigascience/giy065. PMID: 29860392; PMCID: PMC6007545.

[24] Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R, Lanczi L, Gerstner E, Weber MA, Arbel T, Avants BB, Ayache N, Buendia P, Collins DL, Cordier N, Corso JJ, Criminisi A, Das T, Delingette H, Demiralp Ç, Durst CR, Dojat M, Doyle S, Festa J, Forbes F, Geremia E, Glocker B, Golland P, Guo X, Hamamci A, Iftekharuddin KM, Jena R, John NM, Konukoglu E, Lashkari D, Mariz JA, Meier R, Pereira S, Precup D, Price SJ, Raviv TR, Reza SM, Ryan M, Sarikaya D, Schwartz L, Shin HC, Shotton J, Silva CA, Sousa N, Subbanna NK, Szekely G, Taylor TJ, Thomas OM, Tustison NJ, Unal G, Vasseur F, Wintermark M, Ye DH, Zhao L, Zhao B, Zikic D, Prastawa M, Reyes M, Van Leemput K. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Trans Med Imaging. 2015 Oct;34(10):1993-2024. doi: 10.1109/TMI.2014.2377694. Epub 2014 Dec 4. PMID: 25494501; PMCID: PMC4833122.

[25] S. Bakas *et al.*, “Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge,” 2018.

[26] S. Bakas *et al.*, “Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features,” vol. 4, no. 1, p. 170117, 2017, doi: 10.1038/sdata.2017.117. [Online]. Available: https://doi.org/10.1038/sdata.2017.117

[27] A. L. Simpson *et al.*, “A large annotated medical image dataset for the development and evaluation of segmentation algorithms,” 2019.

[28] B. N. Kim, J. Dolz, P.-M. Jodoin, and C. Desrosiers, “Privacy-net: An adversarial approach for identity-obfuscated segmentation of medical images,” vol. 40, no. 7, pp. 1737–1749, 2021.

[29] S. Moqurrab, A. Anjum, U. Manzoor, S. Nefti-meziani, N. Ahmad, and S. Malik, “Differential Average Diversity: An Efficient Privacy Mechanism for Electronic Health Records,” vol. 7, pp. 1177–1187, 2017, doi: 10.1166/jmihi.2017.2146.

[30] P. Mohassel and Y. Zhang, “SecureML: A System for Scalable Privacy-Preserving Machine Learning,” presented at the - 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 19–38, doi: 10.1109/SP.2017.12.

[31] J. Yuan and S. Yu, “Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing,” vol. 25, no. 1. pp. 212–221, 2014.

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” presented at the Artificial intelligence and statistics, 2017, pp. 1273–1282.

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

[34] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” vol. 10, no. 2, pp. 1–19, 2019.

[35] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. L-diversity: Privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data 1, 1 (March 2007), 3–es. https://doi.org/10.1145/1217299.1217302

[36] N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,” presented at the - 2007 IEEE 23rd International Conference on Data Engineering, 2007, pp. 106–115, doi: 10.1109/ICDE.2007.367856.

[37] Xiaokui Xiao and Yufei Tao. 2007. M-invariance: towards privacy preserving re-publication of dynamic datasets. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD '07). Association for Computing Machinery, New York, NY, USA, 689–700. https://doi.org/10.1145/1247480.1247556

[38] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (Feb. 1978), 120–126. https://doi.org/10.1145/359340.359342

[39] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” presented at the Theory of, 2006, pp. 265–284.

[40] Q. Zhang, J. Ma, Y. Xiao, J. Lou and L. Xiong, "Broadening Differential Privacy for Deep Learning Against Model Inversion Attacks," 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 1061-1070, doi: 10.1109/BigData50022.2020.9378274.

[41] K. B.N. and D. Toshniwal, “Privacy Preserving Naïve Bayes Classification Using Trusted Third Party Computation over Distributed Progressive Databases,” presented at the Advances in Computer Science and Information, 2011, pp. 24–32.

[42] Y. Zhang and S. Bai, "An Improved LRP-Based Differential Privacy Preserving Deep Learning Framework," 2021 17th International Conference on Computational Intelligence and Security (CIS), 2021, pp. 484-488, doi: 10.1109/CIS54983.2021.00106.

[43] P. Zhang, Y. Tong, S. Tang, and D. Yang, “Privacy Preserving Naive Bayes Classification,” presented at the Advanced Data Mining and, 2005, pp. 744–752.

[44] Yu Zhu and Lei Liu. 2004. Optimal randomization for privacy preserving data mining. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04). Association for Computing Machinery, New York, NY, USA, 761–766. https://doi.org/10.1145/1014052.1014153

[45] S. Gambs, B. Kégl, and E. Aïmeur, “Privacy-preserving boosting,” vol. 14, no. 1, pp. 131–170, 2007, doi: 10.1007/s10618-006-0051-9. [Online]. Available: https://doi.org/10.1007/s10618-006-0051-9

[46] B. N. Keshavamurthy, M. Sharma, and D. Toshniwal, “Privacy-preserving Naive Bayes classification using trusted third party and different offset computation over distributed databases,” presented at the - 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), 2010, pp. 362–365, doi: 10.1109/PDGC.2010.5679968.

[47] Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD '00). Association for Computing Machinery, New York, NY, USA, 439–450. https://doi.org/10.1145/342009.335438

[48] Tingting Chen and Sheng Zhong. 2009. Privacy-preserving backpropagation neural network learning. Trans. Neur. Netw. 20, 10 (October 2009), 1554–1564. https://doi.org/10.1109/TNN.2009.2026902

[49] Tingting Chen and Sheng Zhong. 2009. Privacy-preserving backpropagation neural network learning. Trans. Neur. Netw. 20, 10 (October 2009), 1554–1564. https://doi.org/10.1109/TNN.2009.2026902

[50] A. C. -C. Yao, "How to generate and exchange secrets," 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), 1986, pp. 162-167, doi: 10.1109/SFCS.1986.25.

[51] Y.-C. Chang and C.-J. Lu, “Oblivious Polynomial Evaluation and Oblivious Neural Learning,” presented at the Advances in Cryptology — ASIACRYPT 20, 2001, pp. 369–384.

[52] M. Barni, C. Orlandi and A. Piva, "A privacy-preserving protocol for neural-network-based computation", *Proc. 8th Workshop Multimedia Security*, pp. 146-51, 2006.

[53] Agarwal R, Evmfimievski A, Srikant R. Information sharing across private databases. *ACM SIGMOD.* 2003. pp. 86–97.

[54] Malin BA, Sweeney L. A secure protocol to distribute unlinkable health data. AMIA Annu Symp Proc. 2005;2005:485-9. PMID: 16779087; PMCID: PMC1560734.

[55] Lingjuan Lyu, Xuanli He, Yee Wei Law, and Marimuthu Palaniswami. 2017. Privacy-Preserving Collaborative Deep Learning with Application to Human Activity Recognition. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). Association for Computing Machinery, New York, NY, USA, 1219–1228. https://doi.org/10.1145/3132847.3132990

[56] P. Li *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," vol. 74, pp. 76–85, 2017, doi: 10.1016/j.future.2017.02.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17302005

[57] A. Boulemtafes, A. Derhab, and Y. Challal, "Privacy-preserving deep learning for pervasive health monitoring: a study of environment requirements and existing solutions adequacy," vol. 12, no. 2, pp. 285–304, 2022, doi: 10.1007/s12553-022-00640-3. [Online]. Available: https://doi.org/10.1007/s12553-022-00640-3

[58] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. Trans. Info. For. Sec. 13, 5 (May 2018), 1333–1345.

[59] Hao, Meng et al. "Towards Efficient and Privacy-Preserving Federated Deep Learning." *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)* (2019): 1-6.

[60] Phong, Le Trieu and Tran Thi Phuong. "Privacy-Preserving Deep Learning for any Activation Function." *ArXiv* abs/1809.03272 (2018): n. pag.

[61] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 1175–1191. https://doi.org/10.1145/3133956.3133982

[62] P. Kairouz *et al.*, "Advances and open problems in federated learning," vol. 14, no. 1–2, pp. 1–210, 2021.