Visual Computing Report

Privacy-first health research with federated learning

By:
Parth Nitesh Thakkar
(CS22MTECH14005)

TA:
Zarka Bashir
Research scholar,
IIT Hyderabad

Guided by:
Prof. C Krishna Mohan

Indian Institute of Technology Hyderabad

Abstract

Healthcare research has become increasingly data-driven in recent years, with researchers using machine learning and other data analysis techniques to develop better models for understanding, diagnosing, and treating various diseases. However, the sensitive nature of medical data makes it difficult to share and aggregate healthcare data across organizations. This is especially challenging in healthcare research, where large datasets are required to conduct meaningful studies.

One approach to addressing this challenge is federated learning, which is a distributed machine learning technique that enables multiple organizations to collaborate on data analysis without sharing the raw data. Federated learning allows each organization to train their machine learning model on their local data, and then aggregate the results to create a global model. This approach ensures that the raw data remains secure and confidential while still allowing organizations to collaborate on data analysis.

However, privacy concerns remain a significant obstacle in federated learning, especially in healthcare research where the data is highly sensitive. Federated differential privacy is an emerging technology that addresses this challenge by adding an extra layer of privacy protection to federated learning. Federated differential privacy ensures that the data from each organization is kept private by adding random noise to the data before it is shared. This approach ensures that only aggregate statistics are shared, while the raw data remains secure and confidential.

Federated differential privacy has the potential to facilitate large-scale healthcare research by enabling collaborative analysis of diverse datasets from multiple sources without compromising individual privacy. This approach can help reduce disparities in healthcare outcomes by enabling the inclusion of more diverse patient populations in research studies. For example, it can help researchers analyze data from different geographical locations, which can provide valuable insights into the effectiveness of different treatments for different populations.

In summary, the need for federated differential privacy in healthcare research is crucial for enabling more effective, efficient, and equitable healthcare solutions. Federated differential privacy can help protect patient privacy while allowing researchers to collaborate on data analysis, which can lead to more accurate and reliable healthcare insights.

Table of Contents

Chapter 1: Introduction

Most health research uses centralized databases that have full access to sensitive data, but recent advances in distributed machine learning allow for building complex models without centralized databases. Federated learning is a type of machine learning where multiple participants collaborate to learn a joint model, allowing for privacy-preserving calculation of study endpoints without sharing raw data. Each client's data remains private, with only focused model updates leaving the client. Communication can be peer-to-peer or involve a central orchestrator.

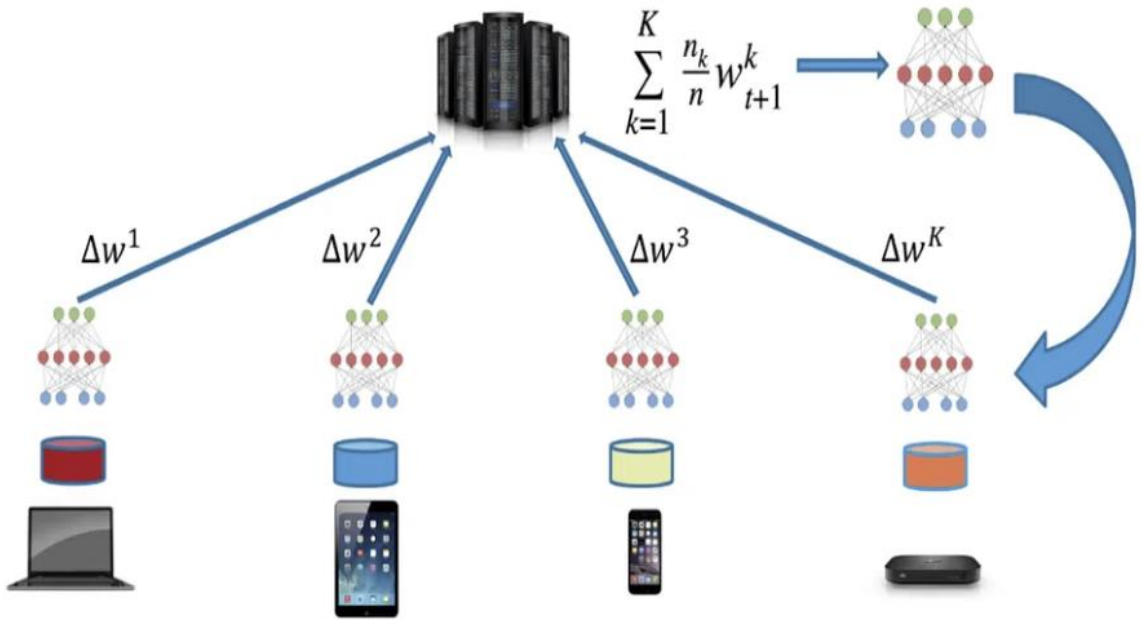


Figure 1:Federated Learning

Federated learning offers two main benefits. First, it allows for a higher quality machine learning model to be developed by leveraging a broader range of data points from multiple participants. This is important for modern models that require large amounts of data. Second, it preserves privacy by keeping participants' sensitive data local and private. Differential privacy is incorporated into the approach to protect individuals' privacy.

Secure aggregation is a technique used in federated learning to protect the privacy of the individual nodes' data while allowing the aggregation of their updates to the shared model.The basic idea of secure aggregation is to encrypt the updates sent by each node before sending them to the central server. This ensures that the data remains private and secure during transmission. Once the encrypted updates arrive at the central server, they are decrypted, aggregated, and then re-encrypted before being sent back to the nodes.

One common method used for secure aggregation is called "homomorphic encryption". Homomorphic encryption allows computations to be performed on encrypted data without revealing the data itself. In the context of federated learning, homomorphic encryption can be used to perform the aggregation of encrypted updates without ever revealing the individual updates themselves.

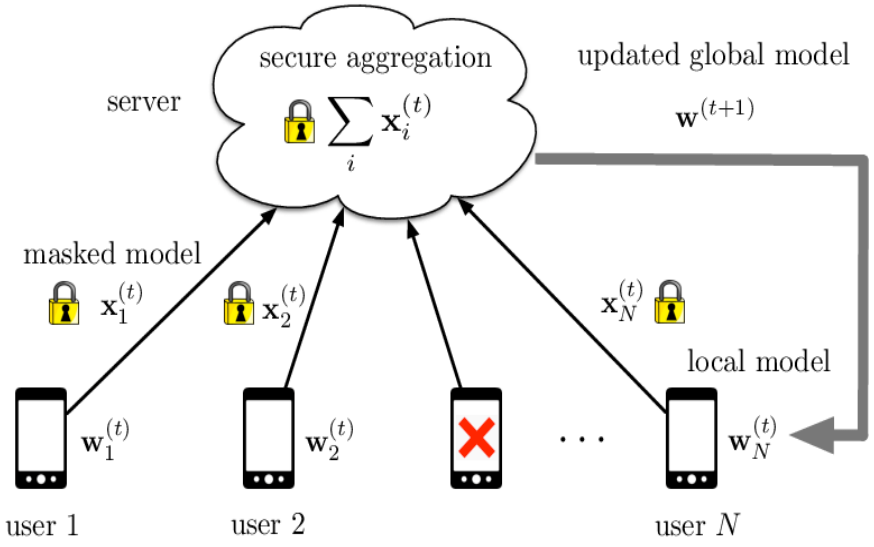


Figure 2:Secure Aggregation

Despite the benefits of federated learning for health research, there are considerable challenges and open questions that have not been systematically studied. Health research often involves a small number of participants with limited data per participant, non-IID data, and potentially unequal levels of participation. Machine learning methods often assume IID data, so it's important to examine the effects of non-IID data in a federated setting. Additionally, prior methods for correlational analysis and hypothesis testing are often ad-hoc, and the generalizability to new datasets is uncertain. Privacy is also a concern in federated learning and needs to be explicitly studied and implemented. The authors propose examining a spectrum of units of the federation and machine learning tasks to address these challenges.

Differential privacy is a technique used to protect the privacy of individuals in datasets. It ensures that an individual's data cannot be linked to their identity, even if an attacker has access to other information. Differential privacy works by adding noise to the data before it is released to the public. This noise is carefully calibrated so that statistical inferences about the data remain accurate, while at the same time ensuring that individual data points are protected. The amount of noise added depends on a parameter called epsilon, which controls the level of privacy protection.

$$\Pr[\mathcal{A}(D_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{A}(D_2) \in S]$$

Figure 3:Definition of ε-differential privacy

Let ε be a positive real number

Let "A" be a randomized algorithm that takes a dataset as input

D1,D2 are the datasets that differ by atmost one entry

S is the set of all subsets that algorithm A will give as an output

Differential privacy has become increasingly important as more and more data is collected and shared. It is widely used in applications such as census data, medical research, and online advertising to protect the privacy of individuals while still allowing for useful analysis to be performed on the data.

Our work demonstrates the successful use of federated learning in the presence of these challenges in homogeneous data silo settings (i.e., where the output of federated computation from one data silo is composable with the output from another silo). Specifically, in this work we reproduce eight diverse health studies in terms of study design, statistical analysis, and sample size, spanning the past several decades in a purely federated setting, where 1 each unit of federation keeps their data private but still contributes to the aggregate model. We randomly sampled seven observational studies and one clinical trial that generated new knowledge on various clinical and epidemiological problems and made the underlying raw data publicly available. The focus of these studies ranges from diabetes to heart disease to SARS-CoV-2 and MERS-CoV to patient mortality prediction based on electronic medical records.

Finally, to test various units of the federation, we experiment with the extreme case of each patient being its own unit, as well as with groups of patients. Four out of the eight datasets are studied at both the individual level of the federation as well as larger generally non-IID units (silos), such as hospital unit, communities, or countries. These groupings were taken from the original data structure to mimic real-world settings and complexities as closely as possible. To explore a broader range of types of silos when the original data does not contain such naturally-occurring silos, we also silo the data randomly using a Dirichlet distribution and run cross-silo experiments on the resulting grouping. Additionally, rather than developing a custom technique to federate learning of one specific class of models as done in prior work, we demonstrate how such encompassing work can be achieved within the unified framework of TensorFlow.

Chapter 2: Literature Review

This paper focuses on federated learning across individual patients' data that can be stored independently. By contrast, most existing applications of federated learning to health research involve several bulk data holders (for example, academic research centers, pharmaceutical companies, or hospitals) collaboratively training models on their entire joint datasets, containing data about many individuals, all at once. The two approaches are termed "cross-device" and "cross-silo" federated learning respectively, and are described in-depth in Kairoz. While this paper focuses on model training via federated learning (FL), federated analytics (FA)—the application of data science techniques to data that is stored locally on client devices—holds similar promise for health research, within the scope of federated analytics lie averages, histograms, heavy-hitter identification, quantiles, set cardinality, covariance matrix estimation, clustering, dimensionality reduction, graph connectivity, and more.

Cross-silo federated learning has already been applied in the healthcare arena to power clinical research among participating hospitals or pharmaceutical companies. In these applications, each participant holds a significant amount of data, sufficient for independent analysis; federated learning improves the quality of this analysis by leveraging data held by multiple participants. By contrast, in this work, we focus on those scenarios commonly found in epidemiological health studies, specific studies with many participants, each of whom has relatively small amounts of non-IID, labelled data. The approach described here can be appropriate for health studies involving smartphone/wearable data and virtual clinical studies (also called decentralized clinical studies) that directly recruit individual research participants without relying on clinical sites for recruitment.

Applications of cross-device federated learning for medical research include: (1) training models on data that is held directly by individuals—for example, health or behavioural data collected on their phones—without requiring a trusted centralized collector, and making use of data signals that are too sensitive or resource-intensive to transmit to a central location. Significant prior work exists evaluating federated learning in the cross-device setting, where many clients each hold their own training examples. Especially when combined with differential privacy, the literature demonstrates privacy gains in these scenarios. Motivation Most health research to date uses data stored in a centralized database (i.e., a database stored in a single site), where analysis and model fitting is done with full access to the sensitive underlying data Federated learning techniques enable calculation of research study endpoints in a privacy-preserving fashion such that private data never leaves a given device. With these advantages, large fractions of the population that wanted to contribute to novel health findings, but had reservations about sharing raw data and digital signals, Now will contribute towards the same without any hesitations.

Chapter 3: Motivation

Health research often requires access to large amounts of sensitive data, such as medical records, genetic information, and lifestyle data. Traditionally, this data has been stored in centralized databases, where researchers can access it for analysis and modeling. However, this approach has raised concerns about privacy and security, as the data is often shared across multiple parties and can be vulnerable to breaches.

Federated learning is a privacy-preserving approach that allows researchers to calculate study endpoints without ever accessing the raw data. Instead, the data remains on the devices of the participants or healthcare providers, and only aggregated information is shared with the research team. This allows for large fractions of the population to contribute to research without having to share their sensitive data.

In federated learning, the process begins with a model being sent to the participants' devices, where it is trained on their local data. The model is then updated with the new information and sent back to the research team, who aggregate the updated models from all the participants. The final model reflects the insights from all participants without revealing their individual data. This approach not only protects privacy but also improves the quality and representativeness of the research findings, as the data is diverse and more reflective of the population.

By using federated learning, individuals can contribute to research studies without having to share their sensitive data with multiple parties, reducing the risk of data breaches and privacy violations. This approach also promotes a more collaborative and participatory approach to research, where individuals can contribute their data while retaining control over it. Overall, federated learning has the potential to revolutionize the way health research is conducted, allowing for more diverse, representative, and privacy-preserving studies.

Chapter 4: Methodology

Our study has shown that federated learning can be successfully implemented inhomogeneous data silo settings, where the output of federated computation from one data silo can be combined with the output from another silo. Unlike previous studies that focused on developing customized techniques for federating the learning of a specific class of models, we have demonstrated how this can be achieved within the unified framework of TensorFlow, which is a more encompassing approach.

We will use Logistic Regression and DNN models for our testing. To use existing datasets in a federated learning setting, we partitioned the original centralized dataset to simulate the data being distributed across research participants. Each participant is treated as an individual client, contributing with varying participation rates to learn a model jointly. This approach works well in settings where each participant captures multiple data examples, and the aggregation happens as part of the local computation. In such settings, each participant can contribute multiple data rows, which reduces the constraints imposed by early aggregation. Our experiments on electronic health records, which consist of complex sequential data spanning a period of hospitalization, demonstrate the effectiveness of this approach. However, it's worth noting that this partitioning approach was only necessary due to the availability of prior studies' datasets. In most cases, existing datasets have already been collapsed to one row of data per participant.

$$z = \frac{1}{1 + exp(-(\beta_0 + \sum_i \beta_i x_i))}$$

Figure 4:Logistic Regression Model

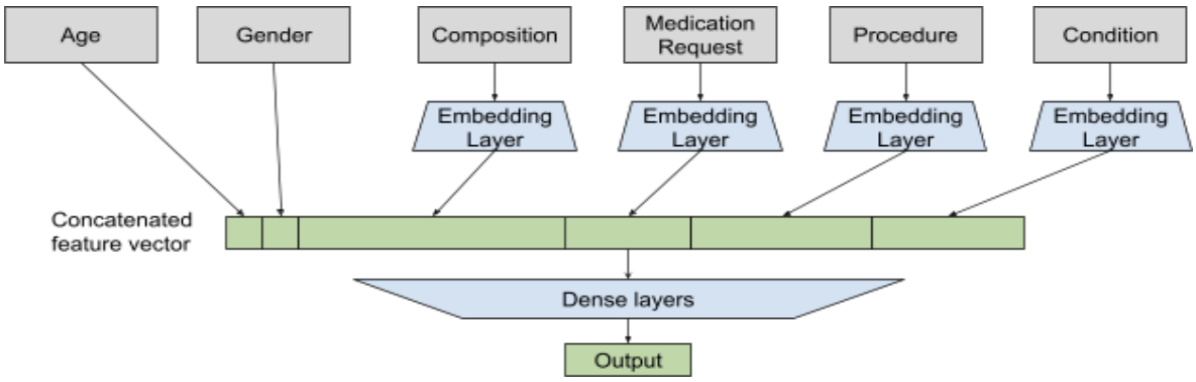


Figure 5:Deep Learning Model

Since in general in the federated setting not all participants may be available at any one time, we explore model quality as a function of subjects' participation rate. Across the datasets, we find that only a minority of clients need to participate in any one round.

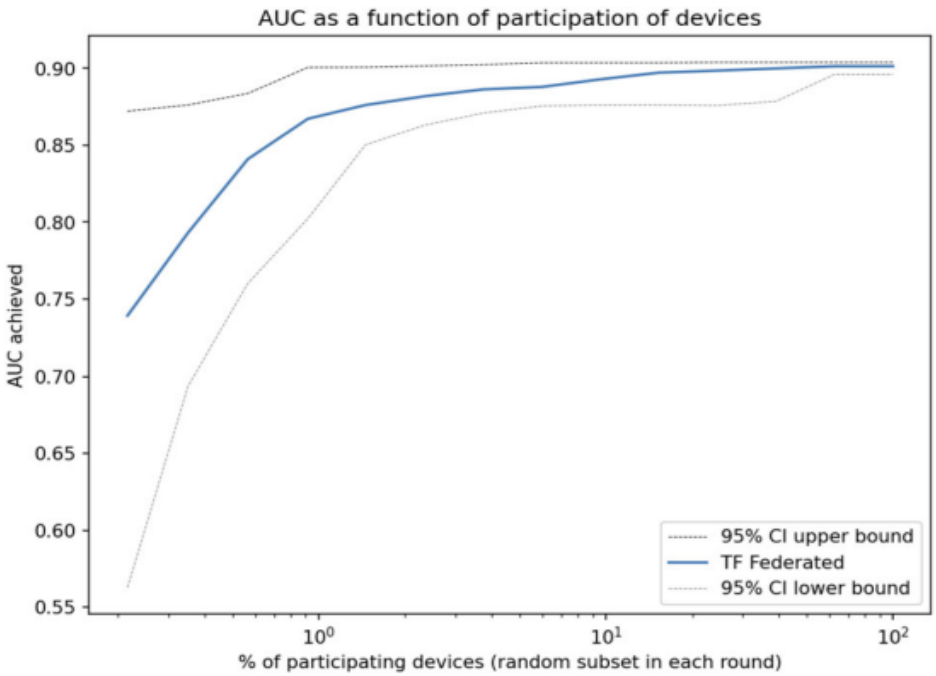


Figure 6:Area under the ROC curve (AUC) as a function of the fraction of participants in each federated (server) round of learning for a replicated model of SARS-CoV-2 and Cancer

Datasets

Table 1. Summary of datasets and methods reproduced in this work.

Study Topic	Manuscript	Study Design	Unit of Analysis	*N*	Statistical Model	Additional Methods	Measure	Covariates
Heart Failure	Chicco & Jurman[11]	Cohort Study	Individual	299	Logistic Regression	N/A	AUC	12
Diabetes	Smith, et al.[27]	Cohort	Individual	768	Neural network with 1 hidden layer	N/A	AUC	8
MIMIC-III	Johnson et al.[12]	Database	Individual & Dirichlet grouping	53,423	Deep neural network	N/A	N/A	92
SARS-CoV-2	Rugge et al.[13]	Cohort Study	Individual & random size grouping	9275	Logistic Regression	N/A	Odds Ratio	3
Avian Influenza	Fiebig et al. [28]	Case Series	Individual & country grouping	294	Logistic Regression	Forward/Backward Selection	Odds Ratio	4
Bacteraemia	Harris et al.[29]	Case Control	Individual	159	Logistic Regression	Forward/Backward Selection	Odds Ratio	12
Azithromycin	Oldenburg et al.[30]	Cluster Randomized Trial	Individual & community of residence grouping	1712	GLM with a Binomial response and the log link	Standard Errors Clustered	Risk Ratio	0
Tuberculosis	Ohene et al.[31]	Case Series	Individual & multi-center grouping	3342	Logistic Regression	N/A	Odds Ratio	3

GLM Generalized Linear Model, *AUC* Area under the Receiver Operating Characteristic Curve.

Figure 7:Summary of datasets used and methods in this work

Heart failure

The Heart Failure Clinical Records Dataset from the University of California Irvine data repository involves 299 individuals with left ventricular systolic dysfunction and New York Heart Association class III or class IV heart failure ranging from 40 to 95 years of age11. The dataset was collected in 2015 from the Faisalabad Institute of Cardiology and the Allied Hospital in Faisalabad in Pakistan. The dataset is used to predict survival, based on 13 attributes including age, sex, blood pressure, left ventricular ejection fraction, diabetes, anemia, and creatinine levels.

Electronic medical records (MIMIC-III)

MIMIC-III is a freely available critical care electronic health records (EHR) database involving comprehensive data from ~40,000 distinct patients age 16 and older, spanning over 53,000 hospital admissions to Beth Israel Deaconess Medical Center between 2001 and 201212. The dataset contains 4579 charted observations and 380 laboratory measurements associated with hospital admis□sions. Each patient in the dataset has a time series of medical encounters involving procedures, medications, diagnoses and other complex signals, such as medical notes.

SARS-CoV-2 and Cancer

Based on the original work, the dataset contains three types of patients: a) Hospitalized, b) ICU admitted, and c) Deceased. For each type of patient, the analysis is divided into two parts based on: i) cancer interval and ii) cancer type. Cancer interval is the number of years a patient suffers from cancer before getting infected by Covid-19. There are different types of cancer reported in the dataset.

Avian influenza A (H5N1)

The Robert Koch Institute (RKI) avian influenza monitoring system is a publicly available epidemiological database established to track avian influenza infections in humans and animals around the world. This database includes 294 human cases from 12 different countries from 2006-2010, and it is used to predict risk of infection and mortality based on country, age, sex, time from symptom onset to hospitalization and exposure to poultry.

Diabetes

The Pima Indians Diabetes Dataset from the Kaggle machine learning data repository is a binary classification database involving females of Pima Indian heritage (https://www.kaggle.com/uciml/pima-indians-diabetes-database). This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases which began long-term longitudinal studies of the onset of diabetes in this population. The dataset is used to predict whether or not a patient will develop diabetes in 5 years time, based on eight attributes including age, body mass index, number of prior pregnancies, blood pressure, insulin and glucose levels.

Bacteraemia

This bacteremia database involves 159 case-controlled cases of bacteraemia occurring among those of age 17 or over at four hospitals in Queensland and New South Wales, Australia between 1998 and 201. 10 The data is used to predict risk factors associated with relapsed infection in patients with Enterobacer bacteraemia, based on multiple factors including age, sex, location, source of infection, hospital location, co-morbid conditions, and many other clinical factors.

Azithromycin in Infants

The Macrolides Oraux pour Réduire les Décès avec un Oeil sur la Résistance (MORDOR) community-randomized study dataset is used to describe adverse events associated with azithromycin use in infants from 30 communities in Niger. The dataset includes 1,712 infants aged 1 to 5 months at time of treatment with azithromycin or placebo between

January 2015 to February 2018. The dataset includes adverse events, age, sex, community and whether there were recent health issues prior to treatment.

Extrapulmonary tuberculosis

This Ghana Extra-pulmonary TB dataset is a medical records database of 3,704 TB patients diagnosed from June 2010 to December 2013 at 11 health facilities in Ghana. 12 The study participants include those 15 years and older with no prior history of TB. The study was conducted to understand the predictors of extrapulmonary TB compared to pulmonary TB such as HIV status and gender. The study also describes factors associated with mortality among patients with extrapulmonary TB (EPTB). The study dataset includes type of infection, health outcomes, age, sex, HIV status, site of infection, type of healthcare facility and year of diagnosis.

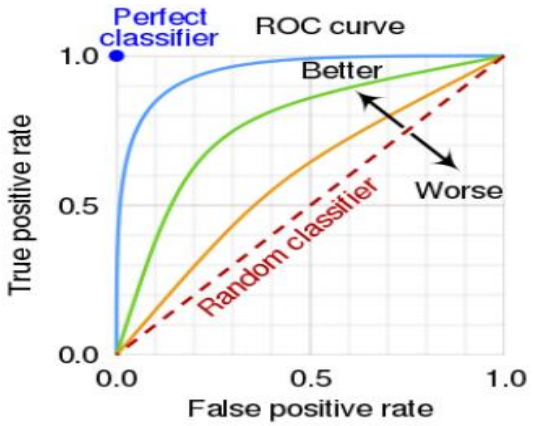
Evaluation Metrics

AUC(Area under the curve)

AUC (Area Under the Curve) is a widely used metric for evaluating the performance of binary classification models. It measures the overall performance of the model across all possible classification thresholds.In a binary classification problem, the model makes predictions on a set of data points, assigning each point a score between 0 and 1, which represents the probability of the point belonging to the positive class. The AUC metric is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold values, and then computing the area under this curve.

The true positive rate is the proportion of actual positive instances that are correctly identified as positive by the model, while the false positive rate is the proportion of actual negative instances that are incorrectly identified as positive by the model.A perfect binary classification model would have an AUC of 1.0, which means that it is able to perfectly distinguish between the positive and negative classes. A model with an AUC of 0.5 is no better than random guessing, and a model with an AUC less than 0.5 is worse than random guessing.

One advantage of the AUC metric is that it is insensitive to class imbalance and threshold selection. This means that it is a good measure of performance even when the classes are imbalanced, and when the decision threshold for classifying positive and negative instances is not well-defined.



Odds Ratio

In machine learning and statistics, odds ratio is a measure of association between two binary variables. It compares the odds of the occurrence of an event in one group to the odds of the occurrence of the same event in another group.More specifically, the odds ratio (OR) is the ratio of the odds of an event occurring in the first group to the odds of the same event occurring in the second group. The odds of an event are defined as the ratio of the number of times the event occurs to the number of times it does not occur.

Mathematically, the odds ratio can be expressed as:

OR = (a/b) / (c/d)

where a is the number of occurrences of the event in the first group, b is the number of non-occurrences of the event in the first group, c is the number of occurrences of the event in the second group, and d is the number of non-occurrences of the event in the second group.

An odds ratio of 1 indicates that there is no association between the two variables, while an odds ratio greater than 1 indicates a positive association (i.e., the occurrence of the event is more likely in the first group than in the second group), and an odds ratio less than 1 indicates a negative association (i.e., the occurrence of the event is less likely in the first group than in the second group).

In machine learning, the odds ratio can be used to evaluate the importance of different features or variables in predicting the occurrence of an event. For example, the odds ratio can be calculated for each feature in a logistic regression model to determine which features are most strongly associated with the outcome variable.

Overall, the odds ratio is a useful measure of association in binary classification problems, and it can be used to evaluate the importance of different features or variables in predicting the occurrence of an event.

Results

Heart failure

The original work presents two logistic regression models—one with all variables and one with only three observed variables (ejection fraction, serum creatinine, and time of follow up in months).

- Our federated setting achieves 0.85 AUC (95% confidence interval of 0.85–0.86) in the full model formulation (cf. 0.82 in the original work) and 0.83 AUC (0.82–0.84) in the latter setup with variable selection (cf. 0.82 in the original work).

The higher AUC score in our setting is due to the addition of regularization while optimizing model parameters, which also allows the new method to subsume the semi-manual variable selection done in the original work. Mirroring the original study, all metrics are reported as means over 100 executions with randomized training, and testing data splits.

- Adding a central differential privacy module (Supplementary Discussion 4) reduces AUC to 0.83 (0.82–0.84) for the full model (cf. 0.82 in the original work which does not consider any DP protections), but provides strong guarantees ($\varepsilon = 0.165$ and $\delta = 10{-}5$).
- With local DP, the federated architecture also achieves 0.83 AUC (0.82–0.84) with local $\varepsilon = 1.36$ and local $\delta = 10{-}9$ per round.

We note this is a very small dataset containing only 299 examples and this experiment demonstrates our methods apply also in situations where data is limited.

Electronic medical records (MIMIC-III)

We build a deep neural network to predict inpatient mortality with data up to 24 h after admission, using patient age, gender, Clinical Classification Software diagnosis codes, RxNorm medication codes, Current Procedural Terminology procedure codes, and free-text medical notes as input variables. The model architecture contains an input layer, three hidden layers with 512, 256, and 128 neurons respectively, and an output layer with a sigmoid activation function (Supplementary Discussion 1, Supplementary Fig. 2). We train the model using the Adam optimizer with a learning rate of 0.01. In addition, and use L1 regularization with magnitude 0.0001 and L2 with 0.01.

To explore different levels of federation, we partition the dataset on a per-patient basis (unit of federation is a single patient) and in groups of patients (per-silo basis). In particular, the per-patient federation follows the cross-device federated learning setting, where each client holds data of a single patient, while the per-silo federation setting splits patients into multiple groups (silos) using a Dirichlet distribution, which simulates the case each hospital or organization holds their patients' data.

To demonstrate the efficacy of federated learning on this dataset, we compare the ROC curve of three different experiments: (1) TF centralized model: A traditional server-side trained model assumes all data is available on a centralized server. (2) TF federated cross-device

model: A model trained on clients on a per patient basis. Each training round has 16 participating patients, and we trained the model for 500 rounds. (3) TF federated crosssilo model: A model training on clients on a per-silo basis. We use a Dirichlet distribution with parameter alpha of 10 to randomly group all patients to 20 groups of various sizes according to the distribution, and select 5 groups at random to participate in each federated training round. The median and interquartile range of patient counts in each group are 200 and 40, respectively.

We measure the performance of three models using the AUC metric and find that all three models achieve comparable performance with greatly overlapping confidence intervals.

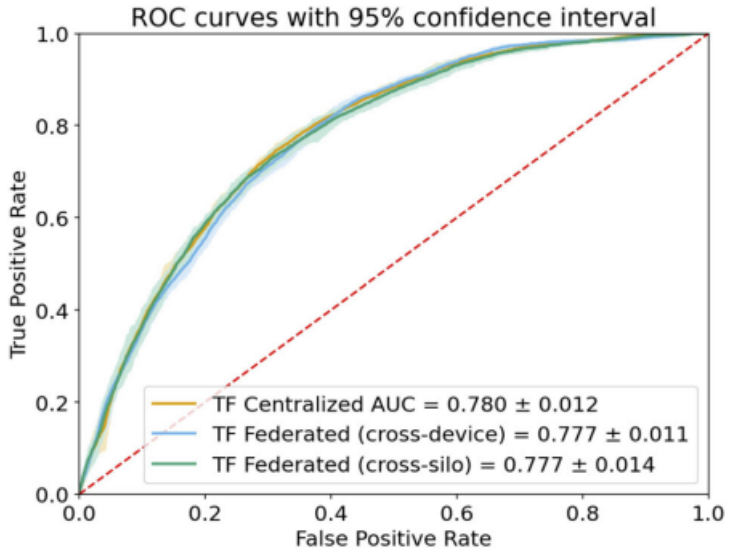


Figure 8: Receiver operating characteristic curves for the three learning setups on MIMIC-III data predicting inpatient mortality. Shaded areas show 95% confidence intervals.

Table 2. Summary of original and federated results reproduced in this work.

Study Topic	Sample Results	Comparison Metric	Traditional Centralized Model[a]	Federated Replications	
				Per-Patient	Per-Silo[b]
Heart Failure	1. Survival Prediction (full model) 2. Survival Prediction (with variable selection)	AUC	0.82 0.82	0.85 0.83	N/A
Diabetes	1. Diabetes prediction at 5-years	AUC	0.84	0.875	N/A
MIMIC-III	1. Inpatient mortality prediction	AUC	0.780± 0.012	0.777 ± 0.011	0.777 ± 0.014
SARS-CoV-2	1. CV2+ve in Female vs. Male 2. CV2+ve in Recent vs. Never Cancer	OR	0.35 (0.32–0.38) 1.88 (1.36–2.60)	0.35 (0.32–0.38) 1.99 (1.45–2.68)	0.35 (0.32–0.38) 2.07 (1.50–2.86)
Avian Influenza	1. Fatality with each day before hospitalization 2. Fatality in Indonesia vs. group of countries	OR	1.33 (1.11–1.60) 0.23 (0.04–1.27)	1.34 (1.12–1.61) 0.25 (0.05–1.37)	1.33 (1.11–1.60) 0.24 (0.04–1.33)
Bacteraemia	1. Relapse with line-associated infection source 2. Relapse with presence of immunosuppression	Coefficient	1.57 (SE: 0.45) 1.07 (SE: 0.41)	1.59 (SE: 0.23) 1.12 (SE: 0.30)	N/A
Azithromycin	1. Adverse events in azithromycin treated	Coefficient	−0.11 (SE: 0.09)	−0.29 (SE: 0.19)	N/A
Tuberculosis	1. Extrapulmonary TB in individuals with HIV	Coefficient	1.16 (SE: 0.09)	1.35 (SE: 0.08)	0.15 (SE: 0.07)[c]

Odds ratios shown as point estimates (95% confidence intervals). Model beta coefficients shown as estimate (standard error).
OR odds ratio, *AUC* Area under the Receiver Operating Characteristic Curve.
[a]As reported in original study or replicated in centralized fashion with statsmodel.
[b]Example silos include hospital level, patient groups, and country level. Not all existing datasets allow meaningful grouping at various levels.
[c]Problem under-specification issue—see additional details in Supplementary Discussion 2 (extrapulmonary tuberculosis).

Figure 9:Summary of all the datasets used and their respective results reproduced in this paper.

Reproduced results(On Heart Failure Dataset)

I have reproduced the results on the Heart failure dataset and I used the Colab environment to run my code, As we can see in Fig 10 The reproduced results are in line with the results mentioned in the paper.

Reproduced results summary	
SKlearn LR AUC :	0.846
TF Centralized LR AUC :	0.846
TF Centralized LR with DP AUC :	0.834
TF Federated LR AUC:	0.840
TF Federated LR with FP AUC:	0.828

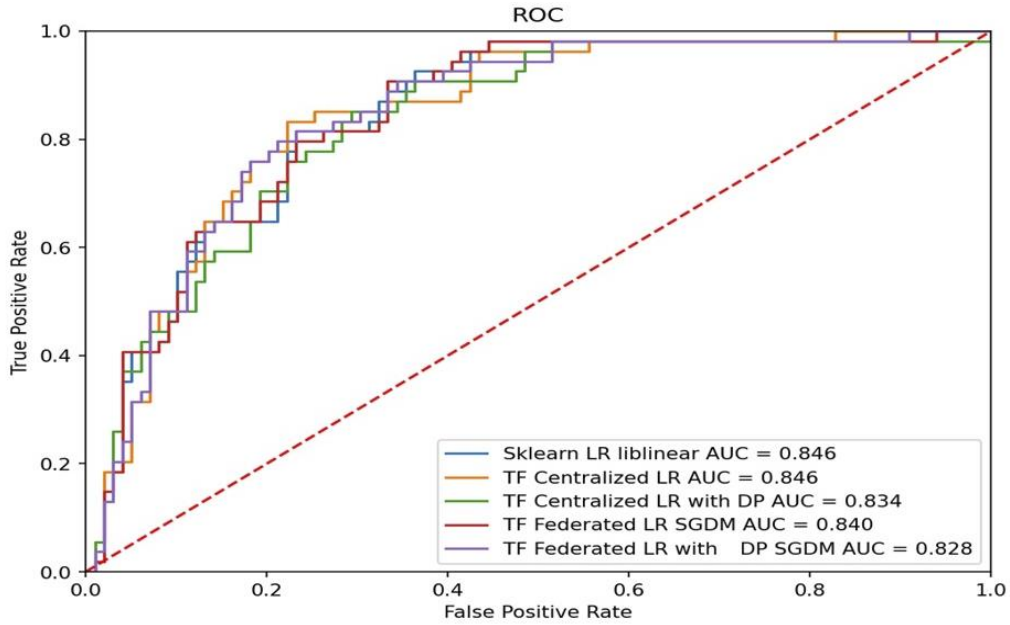


Figure 10:Results obtained while implementing FDP on Heart Failure dataset.

We can observe that we are not losing much of the performance after applying our Federated Differentially private algorithm and we added an extra layer of privacy, so that if the attacker gets hold of the gradients while clients transfer them to the server, he/she will not be able to infer easily the details about the dataset and that was our goal of this paper, some more useful and detailed analysis is performed whose details are mentioned below.

we described that only minority of the clients need to participate in the federated learning round to achieve a considerable amount of performance of our model and also presented the results of the same in the current section Fig 6 which represented Area under the ROC curve (AUC) as a function of the fraction of participants in each federated (server) round of learning for a replicated model of SARS-CoV-2 and Cancer dataset and here I have reproduced the same on Heart failure dataset.

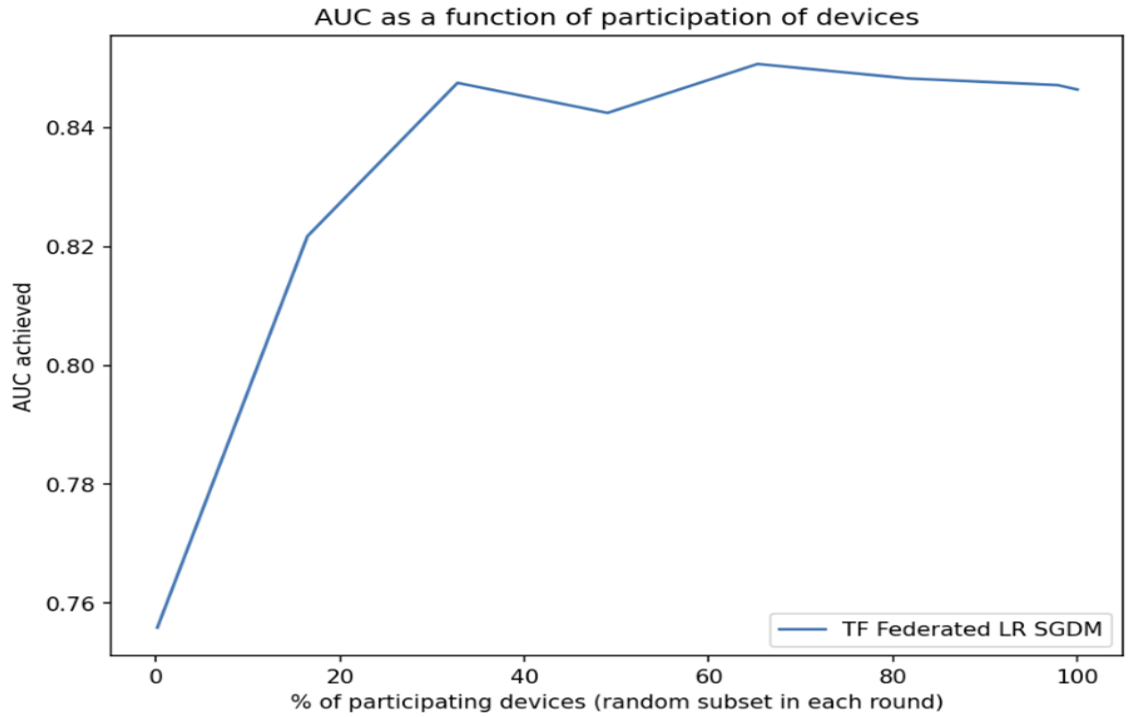


Figure 11: Representing AUC as a function of Percentage of clients participating.

From the above figure(Fig 11) we can clearly see that only about 20-25 percentage of the clients are required to participate in the training round to achieve about 90 percent of the model performance when all the clients are participating, and these results are inline with the results mentioned in the paper.

Now let's analyse the runtime of our FDP model implementation.

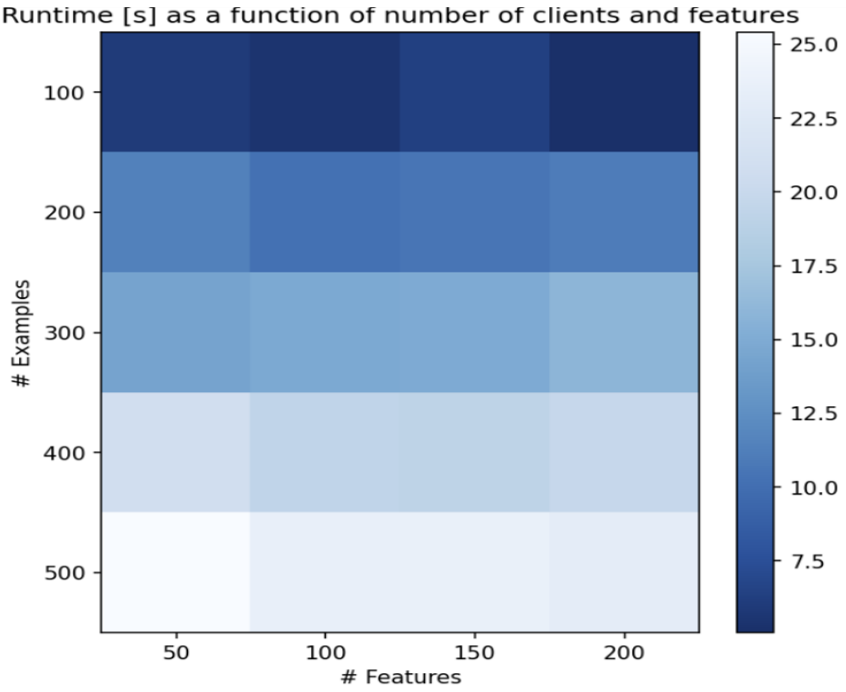


Figure 12:Runtime as a function of clients and features

Figure 12: The runtime of the federated learning process for the Heart Failure problem was represented using shades of blue, and was found to be a function of the number of clients and number of features in each example. The number of clients was equal to the number of examples, since each participant contributed only one example. The results showed that there was a linear relationship between the number of examples/clients and the runtime, indicating that the more clients there were, the longer the process took. However, the dimensionality of the examples did not have a significant effect on the runtime.

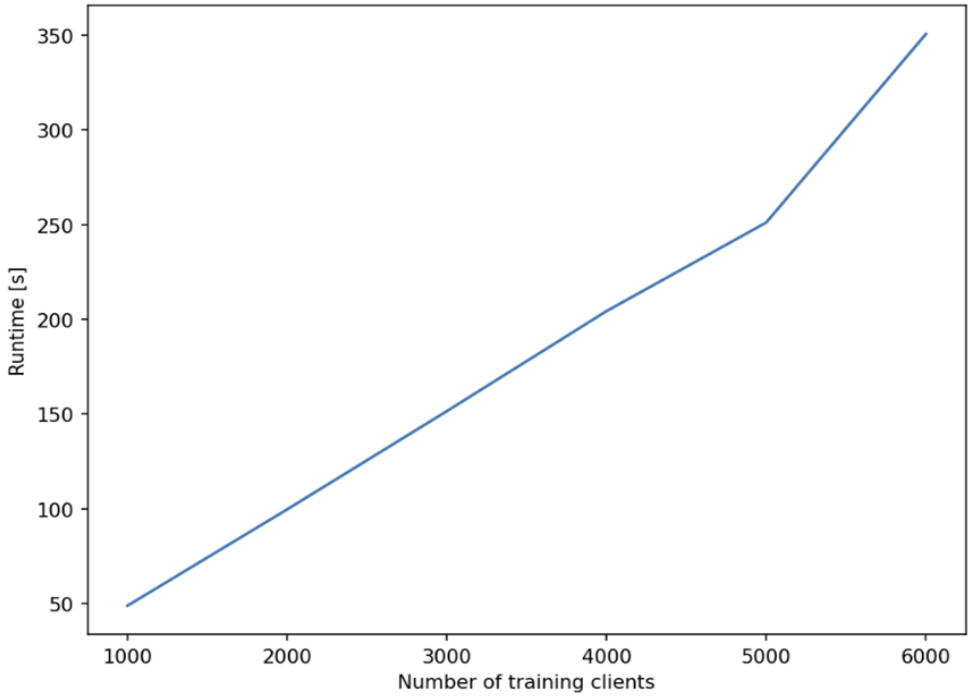


Figure 13:Graph of Runtime vs number of clients participating

Figure 13: We can see a linear relationship between the number of clients and the runtime of our algorithm i.e. as the number of clients are increasing the runtime of our algorithm also increases. But that is always not that bad as number of clients/examples increase the convergence of the model becomes faster and less number of training rounds are re□quired. So, there is always a trade off between the same.

Chapter 5: Novelty

I am thinking of introducing utility-optimized federated learning along with differential privacy in the field of Health care research. I have referred to the paper "Utility Optimization of Federated Learning with Differential Privacy (hindawi.com)" for this idea.

The main challenge of federated learning with differential privacy is to strike a balance between utility and privacy. The smaller value of epsilon will ensure a great privacy-preserving system but, in turn will impact the utility of the system significantly and a bigger value of epsilon will have a less privacy-preserving system but the utility of our system will be great.

What is the issue with Traditional methods used for FDP learning, and what will be our approach to improve upon it?

Differential privacy is a powerful technique used to protect the privacy of individuals when collecting, storing, and analysing their data. In the context of federated learning, differential privacy is used to ensure that the data contributed by each participant remains private while still allowing for the aggregation of the data to improve the overall accuracy of the model.

However, existing differentially private methods allocate the same amount of privacy budget for each participant and each iteration in model updating, which can lead to trade-off problems between privacy and utility. This is because federated learning is a complex training process with many participants and iterative updating, aggregation, and broadcasting. The guarantee of its privacy comes from adding a certain amount of noise at each communication, but this can lead to significant degradation in the accuracy of the model.

To tackle this issue, one approach is to allocate privacy budgets for different participants dynamically. This means that the amount of noise added to the data contributed by each participant is adjusted based on the level of privacy risk associated with their data. For example, participants with more sensitive data may be allocated a larger privacy budget to ensure that their data remains private, while participants with less sensitive data may be allocated a smaller budget.

Dynamic allocation of privacy budgets can also take into account the amount of data contributed by each participant and the number of iterations in the model updating process. This allows for a more fine-grained control of the privacy budget, which can improve the overall accuracy of the model while still protecting the privacy of individual participants.

Overall, dynamically allocating privacy budgets for different participants in federated learning can help to address the trade-off problems between privacy and utility. This approach can lead to more accurate models while still protecting the privacy of individual participants, making federated learning a more practical and effective tool for data analysis in sensitive domains such as healthcare, finance, and social sciences.

Steps of our approach will be:

- **Initialization step**:
  The collaborator initializes the global model parameters such as learning rate, batch size, and noise parameter.
- **Participant selection step**
  The collaborator evaluates available participants based on an optimal indicator and selects participants to join in the current training round.
- **Parameter broadcast step**:
  The collaborator broadcasts the global model and noise parameters of the current round to the selected participants.
- **Local model update:**
  The participants privately and locally perform a round of model training by adding random Gaussian noise to the local weights based on their local data.
- **Local models upload and aggregation:**
  They then upload their private local models to the collaborator for aggregation.
- **Noise scale adjustment:**
  The collaborator verifies the global model convergence by obtaining verification accuracy and decides whether to adjust the noise parameter in the current dynamic adjustment round. If necessary, the collaborator broadcasts the adjusted noise parameter in the next training round.

Input: Dataset $D = \{D_1, \ldots, D_n\}$, initial $\sigma_0 w^{(0)}$, maxRound, $t = 0$, number of client chosen K, p_threshold, learning rate η, batch size B, clipping threshold C, σ_t, max _local_round, noise scale adjusting $k \in (0, 1)$, adjusting threshold α, $S_0 = 0$
Output: Global model weight $w^{(t)}$

(1) $D' \leftarrow D$
(2) $w^{(t)} = w^{(0)}$
(3) $\sigma_t = \sigma_0$
(4) **while** $t <$ max Round **do**
(5) $D' =$ Client choose$(D', K)//2$
(6) Broadcast$(w^{(t)})$
(7) **for** $i \in D', i = 1, \ldots, n$ **do**
(8) $\tilde{w}_i^{(t)} =$ Client_update(D_i, σ_t)//Algorithm 3
(9) Upload$(\tilde{w}_i^{(t)})$
(10) **end**
(11) $w^{(t)} =$ Aggregate$(w_i^{(t)})//i = 1, \ldots, n$
(12) **if** round = dynamic_adjust_round **then**
(13) $\sigma_t =$ Dynamic_adjust(D', σ_t)//Algorithm 4
(14) **end**
(15) $t = t + 1$
(16) **end**
(17) **return** $w^{(t)}$

Figure 14: Algorithm for adaptive DPFL

Figure 14:is our algorithm that summarises all the steps mentioned above and it also has some subroutine algorithms that are described below.

Input: D, number of participant selected K, $p_$threshold
Output: Clients chosen D'
(1) **for** client in client list **do**
(2) score = evaluate client(D)
(3) score list.add(score)
(4) **end**
(5) $p_$list = calculate_p(score list)
(6) sort(p_list)
(7) calculate $p_$threshold(K)
(8) **for** p in p_list **do**
(9) **if** $p \geq p_$threshold **then**
(10) D'add (client)
(11) **end**
(12) **end**
(13) **return** D'

Figure 15:Algorithm for choosing client.

Figure 15: Represents the algorithm used by the collaborator for choosing the clients that will participate in the current round of FL, this is used to defend our model from attacks like communication attacks or poisoning attacks where an adversary can send fake/manipulated updates to the server which will result in degraded performance.

This algorithm will try to minimize the risk of these attacks by choosing clients based on some evaluation. For example their accuracy on a publicly available dataset and if the accuracy is below a certain threshold then do not include the particular client for the current round of FL.

Input: $D_i (i = 1, \ldots, n)$, learning rate η, batch size B, clipping threshold C, σ_t, max_local_round
Output: Private local weight $\widetilde{w}_i^{(t)}$
(1) Local round $j = 0$
(2) **while** $j \leq$ max_local_round **do**
(3) Forward pass(B_j)
(4) $g_t(B_i) \leftarrow \nabla_{w_i^{(t)}} L(w_i^{(t)}, B_i)$//Compute gradient
(5) $\widetilde{g_t} \leftarrow (1/B)(\sum_i g_t(B_i) + N(0, \sigma_t^2 C^2))$//Add noise
(6) $\widehat{g_t}(B_i) \leftarrow \widetilde{g_t}/\max(1, (\|\widetilde{g_t}\|_2/C))$//Clip gradient
(7) $\widetilde{w_i}^{(t)} \leftarrow w_i^{(t)} - \eta \widehat{g_t}$//Apply gradient
(8) j++
(9) **end**
(10) **return** $\widetilde{w}_i^{(t)}$

Figure 16:Algorithm for client update

Figure 16: Represents the algorithm used by the clients to compute the gradients and add noise to the gradients according to the budget provided and then clip the gradients if necessary to avoid exploding gradients problem.

Input: $D, w^{(t)}$, σ_t, noise adjusting scale $k \in (0, 1)$, adjusting threshold α, $S_0 = 0$
Output: Adjusted noise parameters σ_t
(1) $S_{t-1} = S_0$
(2) $S_t = \text{Validate}(D, w^{(t)})$
(3) **if** $S_t - S_{t-1} \leq \alpha$ **then**
(4) $\quad \sigma_t = k\sigma_t$
(5) **else**
(6) $\quad \sigma_t = \sigma_t$
(7) **end**
(8) $S_t = S_0$
(9) **return** σ_t

Figure 17:Algorithm for Dynamic adjust.

Figure 17: Represents the last and the most important algorithm of our idea i.e. dynamic adjust algorithm that will dynamically adjust the the noise parameter according to the convergence of the model.

As the model starts to converge the difference between the accuracy of the last dynamic adjust round and current dynamic adjust round decreases and as soon as this difference is less than a certain threshold, we will update our noise parameter.

Noise parameter = Noise parameter * K, where K is a number between (0,1).

Chapter 6: Conclusion and Future Work

Federated learning is a promising approach for machine learning on distributed data, as it enables multiple parties to collaboratively train a model while keeping their data private. However, the sharing of data can also lead to privacy concerns. To address this, methods like differential privacy and secure aggregation can be used in conjunction with federated learning to ensure better privacy without significant loss of performance. Differential privacy is a method that adds random noise to the data before sharing it, ensuring that individual data points cannot be traced back to specific users. Secure aggregation ensures that the data from each participant is kept private during the aggregation process.

Regularization techniques can also be applied to the model to further enhance privacy. These techniques help to prevent overfitting of the model, which can lead to leakage of sensitive information from the data.

Adaptive differentially private federated learning is a method that strikes a balance between privacy and utility by adjusting the amount of noise added to the data based on the sensitivity of the data. This ensures that the privacy budget is allocated more efficiently and can reduce the difference between federated learning with and without differential privacy. By reducing the trade-off between privacy and utility, adaptive differential privacy can enable more effective collaborative machine learning while preserving privacy.

And in Future we can collaborate FL with CL(Continual Learning) to further increase scope of research in the field of health care, which can be very useful in the case where there is an unexpected breakout of a disease like corona virus in 2020, by combining FL and CL we would be able to train the model on diseases which are faced by humans for the first time and try to find patterns between the diseases on which our model is already trained on.

References

[1] Zhu, W., Kairouz, P., Sun, H., McMahan, B. & Li, W. Federated heavy hitters with differential privacy. Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. PMLR 108, 3837–3847 (2020).

[2] Choudhury, O. et al. Predicting adverse drug reactions on distributed health data using federated learning. AMIA Annu. Symp. Proc. 2019, 313–322 (2020). eCollection 2019.

[3] Johnson, A. E. W. et al. MIMIC-III, a freely accessible critical care database. Sci. Data 3, 160035 (2016)

[4] Bonawitz K. et al. TensorFlow federated: machine learning on decentralized data. (2020). https://www.tensorflow.org/federated (accessed Nov 2020)

[5] Bonawitz K. et al. Practical secure aggregation for privacy preserving machine learning. In Proceedings of ACM Conference on Computer and Communications Security (ACM CCS). (2017).

[6] Video to understand Differential Privacy:-(160) Differential Privacy - Simply Explained - YouTube

[7] R. C. Geyer, T. Klein, and M. Nabi, “Differentially private federated learning: a client level perspective,” 2017, http:// arxiv.org/abs/1712.07557.

[8]Jianzhe Zhao, Keming Mao, Chenxi Huang, Yuyang Zeng, "Utility Optimization of Federated Learning with Differential Privacy", *Discrete Dynamics in Nature and Society*, vol. 2021, Article ID 3344862, 14 pages, 2021. https://doi.org/10.1155/2021/3344862

[9] Geyer R. C., Klein T., Nabi M. Differentially private federated learning: a client level perspective. arXiv 2017; published online Dec. http://arxiv.org/abs/1712.07557 (Accessed 23 Nov 2020).[10] Choudhury, O. et al. Predicting adverse drug reactions on distributed health data using federated learning. AMIA Annu. Symp. Proc. 2019, 313–322 (2020). eCollection 2019.

[10] Ramage D. & Mazzocchi S. Federated analytics: collaborative data science without data collection. Google AI Blog. (2020). https://ai.googleblog.com/2020/05/ federated-analytics-collaborative-data.html (Accessed Nov 2020).[4] Bonawitz K. et al. TensorFlow federated: machine learning on decentralized data. (2020). https://www.tensorflow.org/federated (accessed Nov 2020)