A MINI-PROJECT REPORT

ON

“When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design”

BY

Mohit Sharma
Sanchita Shirur
Anurag Singh

Under the guidance of

Internal Guide
Prof. Preeti Satao

MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY
Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

Department of Computer Engineering

University of Mumbai

May - 2021

MANJARA CHARITABLE TRUST

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

C E R T I F I C A T E

Department of Computer Engineering

This is to certify that

1.Mohit Sharma
2. Sanchita Shirur
3.Anurag Singh

Have satisfactory completed this project entitled

"When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design"

Towards the partial fulfilment of the

THIRD YEAR BACHELOR OF ENGINEERING
IN
(COMPUTER ENGINEERING)

as laid by University of Mumbai.

Guide
Prof Preeti Satao

H.O.D.
Prof. S. P. Khachane

Principal

Dr. Sanjay Bokade

Project Report Approval for T. E.

This project report entitled ”***When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design***” by **Mohit Sharma, Sanchita Shirur & Anurag Singh** is approved for the degree of Third-Year Bachelor of Computer Engineering.

Examiners:

1----------

2.----------

Date: 14/05/2021
Place: Mumbai

Declaration

We wish to state that the work embodied in this project titled **"When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design"** forms our own contribution to the work carried out under the guidance of "Prof Preeti Satao" at the Rajiv Gandhi Institute of Technology.

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Students Signatures)

Mohit Sharma (B-635)

Sanchita Shirur (B-643)

Anurag Singh (B-645)

Abstract

Machine learning models trained on sensitive real-world data promise improvements to everything from medical screening to disease outbreak discovery. And the widespread use of mobile devices means even richer and more sensitive data is becoming available. However, traditional machine learning involves a data pipeline that uses a central server(on-premise or cloud) that hosts the trained model to make predictions. Distributed Machine Learning (FL) in contrast, is an approach that downloads the current model and computes an updated model at the device itself (also known as edge computing) using local data. Federated learning (FL) is a machine learning setting where many clients (e.g. mobile devices or whole organizations) collaboratively train a model under the orchestration of a central server (e.g. service provider) while keeping the training data decentralized.

Most of the previous research-work in federated learning focuses on the transfer and aggregation of the gradients for learning on linear models and very less work is available on non-linear models. In this project, we explore a secure decentralized learning model using neural networks. The motivation for the same came from blockchain, preserving the user identities without a trusted central server and the hidden layers in a neural network, able to add non-linearity to the model. Analogous to the transfer of gradients in the federated learning system which requires a lot of network communication, we explore the possibility of broadcasting weights to the blockchain system.

The goals of this work are to highlight research problems that are of significant theoretical and practical interest and to encourage research on problems that could have significant real-world impact.

Contents

List of Figures		vii
List of Tables		viii
List if Algorithms		ix
1	Introduction IntroductionDescription...…….. Organization of Report…….…….	1 1 2
2	Literature Review 2.1 Survey Existing system…….. 2.2 Limitation Existing system or research gap…… 2.3 Problem Statement and Objective……...	3 3 4 5
3	Proposed System 3.1 Analysis/ Framework/ Algorithm…… 3.2 Details of Hardware & Software……. 3.2.1 Hardware Requirement..…….. 3.2.2 Software Requirement…… 3.3 Design Details……. 3.3.1 System Flow/System Architecture…… 3.3.2 Detailed Design(UML) 3.4 Methodology/Procedures(Your methodology to solve the problem)…..	15 15 17 17 17 18 18 18 18 18
4	Results & Discussions 4.1 Results……. 4.2 Discussion-Comparative study/ Analysis……..	19 19 20
5	Conclusion and Future Work	21
	References	22

LIST OF FIGURES

Figure No.	Name	Page no.
1.1.1	Decentralized System	10
1.1.2	Federated Learning - Demonstration	10
1.1.3	Federated Learning - Working	11
1.1.4	Blockchain Demonstration	11
3.1.1	Linear Regression Model	15
3.1.2	Linear Regression Model - Graphical representation	16
3.1.3	Proof of Work Algorithm	17
3.1.4	Proof of Work-Block structure	17
3.1.5	SHA - Hashing Algorithm	18
3.1.6	The eight input blocks of the initial has value H, in hexadecimal notation	19
3.1.7	Node snippet for producing hexadecimal values	19
3.3.1	System Architecture	20
3.3.2.1	Detailed Design	21
4.1.1	Output - Client 1	22
4.1.2	Output - Server	22
4.1.3	Output	23
4.2.1	Discussion - graph	23

LIST OF ALGORITHMS

Sr. No.	Name	Page no
1	Linear Regression	15
2	Proof of Work	17
3	Secure Hash Algorithm (SHA)	18

CHAPTER 1

Introduction

1.1 Introduction Description

Advancements in the power of machine learning have brought with them major data privacy concerns. This is especially true when it comes to training machine learning models with data obtained from the interaction of users with devices such as smartphones.

The traditional process for training a machine learning model involves uploading data to a server and using that to train models. This way of training works just fine as long as the privacy of the data is not a concern. However, when it comes to training machine learning models where personally identifiable data is involved (on-device, or in industries with particularly sensitive data like healthcare), this approach becomes unsuitable.

Training models on a centralized server also means that you need enormous amounts of storage space, as well as world-class security to avoid data breaches. But imagine if we were able to train your models with data that's locally stored on a user's device.

Federated learning is a model training technique that enables devices to learn collaboratively from a shared model. The shared model is first trained on a server using proxy data. Each device then downloads the model and improves it using data — federated data — from the device.

The device trains the model with the locally available data. The changes made to the model are summarized as an update that is then sent to the cloud. The training data and individual updates remain on the device. In order to ensure faster uploads of these updates, the model is compressed using random rotations and quantization. When the devices send their specific models to the server, the models are averaged to obtain a single combined model. This is done for several iterations until a high-quality model is obtained .

Training the model on client devices using the federated averaging algorithm is shown to perform better than server-based training using stochastic gradient descent. The algorithm is used on the server to combine updates from the clients and produce a new global model. In this project, we explore a secure decentralized learning model using neural networks. The motivation for the same came from blockchain, preserving the user identities without a trusted central server and the hidden layers in a neural network, able to add non-linearity to the model. Analogous to the transfer of gradients in the federated learning system which requires a lot of network communication, we explore the possibility of broadcasting weights to the blockchain system.

Hence, in this work we have discussed the implementation of Federated Learning and Blockchain for training machine learning models using a decentralized approach thereby attempting to protect users' sensitive data .

Decentralization:

A decentralized system is an interconnected information system where no single entity is the sole authority. In the context of computing and information technology, decentralized systems usually take the form of networked computers. For example, the Internet is a decentralized system, although it has become increasingly centralized over time.

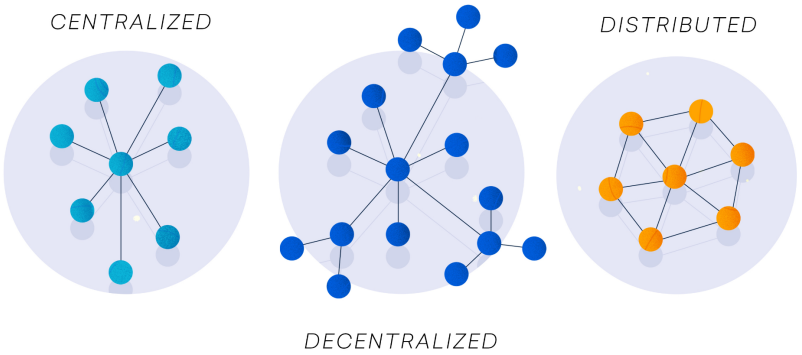


fig1.1.1 Decentralized System

For businesses and organizations, the benefits of a decentralized system include failure tolerance and redundancy. A decentralized system is distinct from a distributed system. A decentralized system generally has multiple authoritative nodes, each of which serves a subset of the total end users. In a distributed system, however, there are no end users, because every node on the network communicates with every other to behave as a single unit.

Decentralized systems have gained widespread attention with the advent of blockchain technologies, such as those used in Bitcoin and other cryptocurrencies.

Federated Learning:

Federated Learning enables mobile phones to collaboratively learn a shared prediction model while keeping all the training data on device, decoupling the ability to do machine learning from the need to store the data in the cloud. This goes beyond the use of local models that make predictions on mobile devices (like the Mobile Vision API and On-Device Smart Reply) by bringing model training to the device as well.

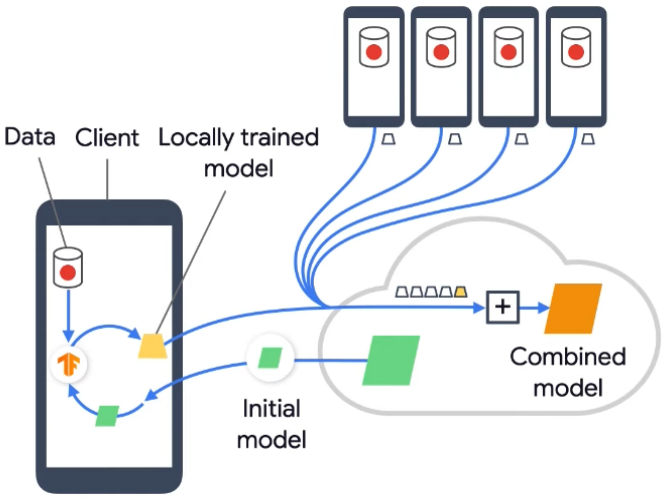


Fig.1.1.2 Federated Learning - Demonstration

Working: Your device downloads the current model, improves it by learning from data on your phone, and then summarizes the changes as a small focused update. Only this update to the model is sent to the cloud, using encrypted communication, where it is immediately averaged with other user updates to improve the shared model. All the training data remains on your device, and no individual updates are stored in the cloud.

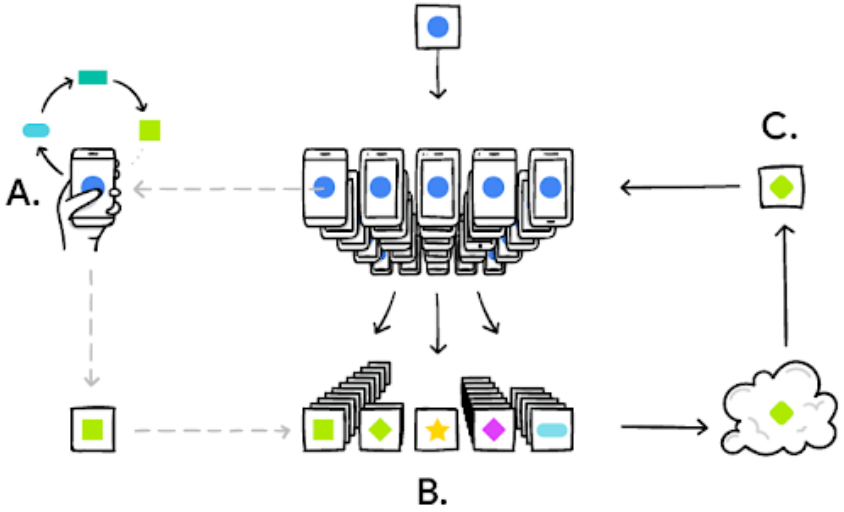


Fig.1.1.3 Federated Learning - Working

Federated Learning allows for smarter models, lower latency, and less power consumption, all while ensuring privacy. And this approach has another immediate benefit: in addition to providing an update to the shared model, the improved model on your phone can also be used immediately, powering experiences personalized by the way you use your phone.

Blockchain:

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system.

A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralised database managed by multiple participants is known as Distributed Ledger Technology (DLT).

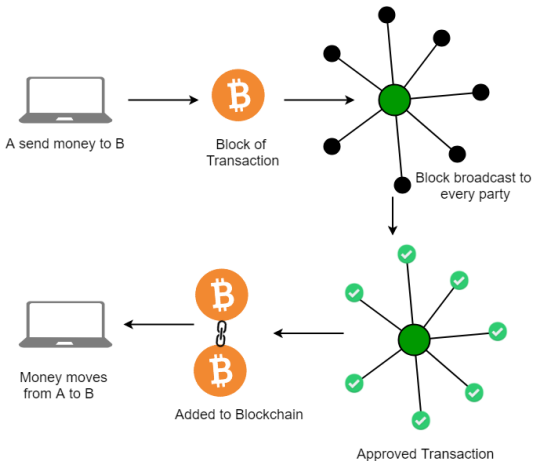


fig. 1.1.4 Blockchain Demonstration

1.2 Organization of report

Describe every chapter (what every chapter contain)

- Ch.1 Introduction: A brief introduction to application of federated learning and blockchain to decentralizing machine learning models to ensure security and data privacy of the user

- Ch.2 Literature Review: Studying the existing systems and limitations that are a threat to users security and data privacy

- Ch.3 Proposed System: Analyzing discrepancies in existing centralized machine learning systems and applying FL and blockchain to overcome these discrepancies.

- Ch.4 Results & Discussion: Using decentralization offered by FL and blockchain, machine learning models can be trained on client side thereby avoiding data breaches and identity thefts

CHAPTER 2

Literature Review

2.1 Survey existing system

Federated learning (FL) has been proposed to allow collaborative training of machine learning (ML) models among multiple parties where each party can keep its data private. FedV, a framework for secure gradient computation in vertical settings for several widely used ML models such as linear models, logistic regression, and support vector machines. FedV removes the need for peer-to-peer communication among parties by using functional encryption schemes; this allows FedV to achieve faster training times[1].
We empirically demonstrate the applicability for multiple types of ML models and show a reduction of 10%-70% of training time and 80% to 90% in data transfer with respect to the state-of-the-art approaches[2].

With the vast growth in the big data era and ever increasing power of mobile computing devices, it's the dire need to build robust distributed learning models. Traditional master worker type of distributed learning assumes a trusted central server and focuses on the privacy issues of the linear learning models[3].

Analogous to the transfer of gradients in the federated learning system which requires a lot of network communication, we explore the possibility of broadcasting weights to the blockchain system.
citation is the serial number of the technical paper which you listed in the reference part.

2.2 Limitation existing system or Research gap

The next generation of artificial intelligence is built upon the core idea revolving around "data privacy". When data privacy is a major concern and we don't trust anyone withholding our data we can turn to federated learning for building privacy-preserving AI by building intelligent systems privately.

Federated learning (FL) is a machine learning setting where many clients (e.g. mobile devices or whole organizations) collaboratively train a model under the orchestration of a central server (e.g. service provider), while keeping the training data decentralized. FL embodies the principles of focused data collection and minimization, and can mitigate many of the systemic privacy risks and costs resulting from traditional, centralized machine learning and data science approaches.

2.3 Problem Statement and Objectives

When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design.
Federated learning is still a relatively new field with many research opportunities for making privacy-preserving AI better. This includes challenges such as system heterogeneity, statistical heterogeneity, privacy concerns, and communication efficiency, etc.. This brings forth many open problems in Federated learning that need to be addressed as a whole before Federated learning can be widely adopted by the industry.

2.3.1 Objectives

Through this project, we aim to implement a secure decentralized learning model using neural networks. Primarily, we aim at developing a Globally Shared Model to where data is and having training models for each user, which eliminates the risk of data leakage and ensures privacy to both parties. Applying this, we can use Machine Learning without putting at risk any of the users personal data. We explore the blockchain technique to propose a decentralized privacy-preserving and secure machine learning system, called LearningChain, by considering a general (linear or nonlinear) learning model and without a trusted central server.

Most of the previous research-work in federated learning focuses on the transfer and aggregation of the gradients for learning on linear models and very less work is available on nonlinear models. In this paper, we explore a secure decentralized learning model using neural networks. Motivation for the same came from the blockchains preserving the user identities without a trusted central server and the hidden layers in the neural network, able to add non-linearity to the model. Analogous to the transfer of gradients in the federated learning system which requires a lot of network communication, we explore the possibility of broadcasting weights to the blockchain system

2.4 Scope

It enables multiple actors to build a common, robust machine learning model without sharing data, thus allowing to address critical issues such as data privacy, data security, data access rights and access to heterogeneous data. Federated learning's applications are spread over a number of industries including defense, telecommunications, IoT, and pharmaceutics.

CHAPTER 3

Proposed System

3.1 Analysis/Framework/Algorithm

Federated Learning enables mobile phones to collaboratively learn a shared prediction model while keeping all the training data on device, decoupling the ability to do machine learning from the need to store the data in the cloud. This goes beyond the use of local models that make predictions on mobile devices by bringing model training to the device as well.

We explore the blockchain technique to propose a decentralized privacy-preserving and secure machine learning system, called LearningChain, by considering a general (linear or nonlinear) learning model and without a trusted central server.

Specifically, we design a decentralized Stochastic Gradient Descent (SGD) algorithm to learn a general predictive model over the blockchain. In decentralized SGD, we develop differential privacy based schemes to protect each party's data privacy and integrity, and propose an l-nearest aggregation algorithm to protect the system from potential attacks. Thus on application of federated and Regression, amalgamation with Blockchain will potentially ensure privacy in a decentralized system.

Algorithm Used -

- Linear regression: Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric. For example, in a simple regression problem (a single x and a single y), the form of the model would be:

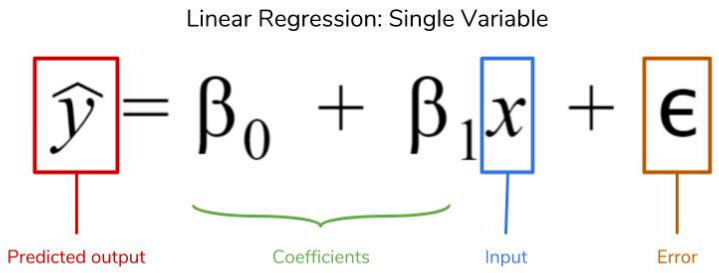


fig. 3.1.1 Linear Regression Model

Usage - It will be used in the Training Algorithm on the client side to perform analysis on the data received from the Particular Client and the aggregate weights.

Steps -

1. Draw the scatter plot. Look for 1) linear or non-linear pattern of the data and 2) deviations from the pattern (outliers). If the pattern is non-linear, consider a transformation. If there are outliers, you may consider removing them only IF there is a non-statistical reason to do so. (Are those individuals "different" than the rest of the sampled individuals?)
2. Fit the least-squares regression line to the data and check the assumptions of the model by looking at the Residual Plot (for constant standard deviation assumption) and normal probability plot (for normality assumption). If the assumptions of the model appear not to be met, a transformation may be necessary.

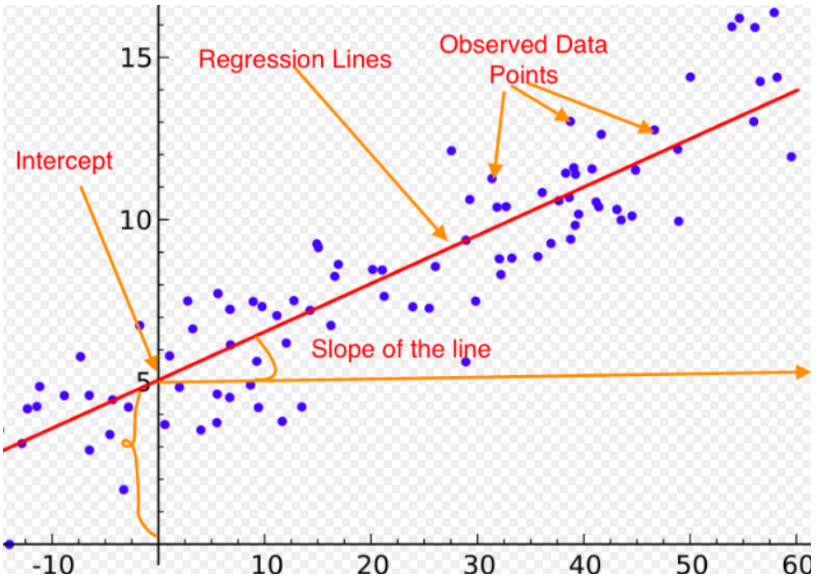


fig. 3.1.2 Linear Regression Model - Graphical representation

3. If necessary, transform the data and re-fit the least-squares regression line using the transformed data.
4. If a transformation was done, go back to step 1. Otherwise, proceed to step 5.
5. Once a "good-fitting" model is determined, write the equation of the least-squares regression line. Include the standard errors of the estimates, the estimate of σ, and R-squared.
6. Determine if the explanatory variable is a significant predictor of the response variable by performing a t-test or F-test. Include a confidence interval for the estimate of the regression coefficient (slope).

Estimation of Error -

We can calculate an error for our predictions called the Root Mean Squared Error or RMSE.

$$RMSE = sqrt(sum((pi – yi)^2)/n)$$

Where sqrt() is the square root function, p is the predicted value and y is the actual value, i is the index for a specific instance, n is the number of predictions, because we must calculate the error across all predicted values.

- Proof of Work(PoW): is a form of cryptographic zero-knowledge proof in which one party (the prover) proves to others (the verifiers) that a certain amount of computational effort has been expended for some purpose. Verifiers can subsequently confirm this expenditure with minimal effort on their part. PoW requires nodes on a network to provide evidence that they have expended computational power (i.e. work) in order to achieve consensus in a decentralized manner and to prevent bad actors from overtaking the network.

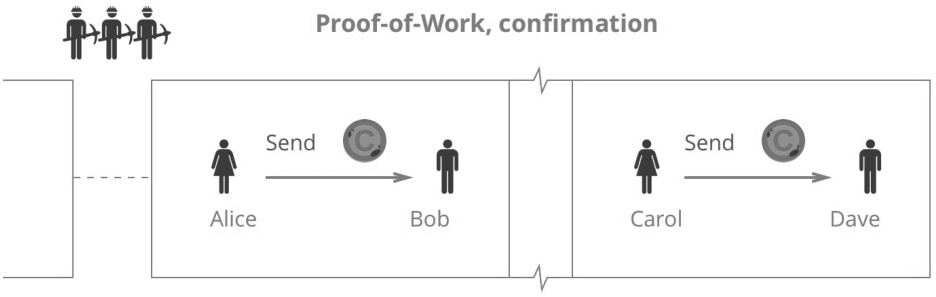


Fig. 3.1.3 Proof of Work Algorithm

Usage - We will use PoW algorithm to confirm the transaction of weights when sent to the Server before Aggregation so that the security is increased and prevents Model Poisoning.

Steps - The process of verifying the transactions in the block to be added, organizing these transactions in a chronological order in the block and announcing the newly mined block to the entire network does not take much energy and time. The energy consuming part is solving the 'hard mathematical problem' to link the new block to the last block in the valid blockchain.

When a miner finally finds the right solution, the node broadcasts it to the whole network at the same time, receiving a cryptocurrency prize (the reward) provided by the PoW protocol. At the time of writing this article, mining a block in the bitcoin network gives the winning miner 12.5 bitcoins. The amount of bitcoins won halves every four years or so(that's how the bitcoin network is designed). So, the next deduction in the amount of bitcoin is due at around 2020-21(with the current rate and growth).

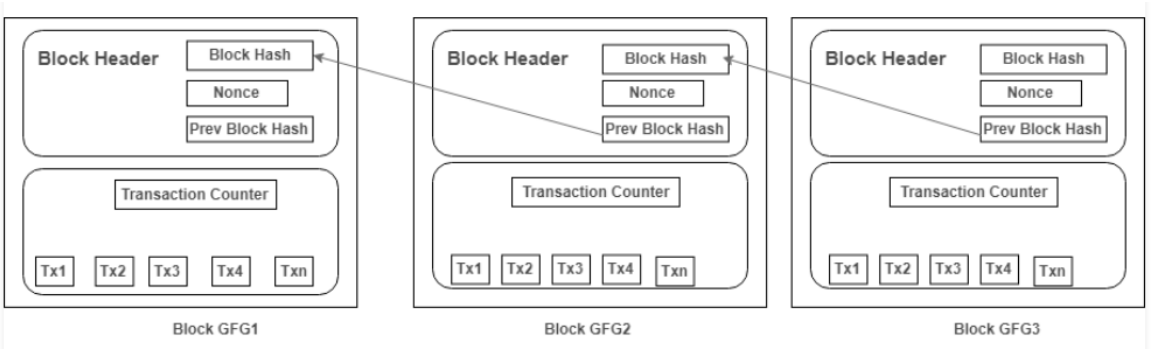


fig. 3.1.4 Proof of Work - Block structure

With more miners comes the inevitability of the time it takes to mine the new block getting shorter. This means that the new blocks are found faster. In order to consistently find 1 block every 10 minutes (That is the amount of time that the bitcoin developers think is necessary for a steady and diminishing

flow of new coins until the maximum number of 21 million is reached (expected some time with the current rate in around 2140)), the Bitcoin network regularly changes the difficulty level of mining a new block.

- SHA 256: In simple words, SHA-256 (Secure Hash Algorithm, FIPS 182-2), is one of the cryptographic hash functions which has a digest length of 256 bits. It's a keyless hash function, meaning an MDC (Manipulation Detection Code). In other words, SHA (Secure Hash Algorithm) was developed by the National Institute of Standards & Technology, and further, they came with a new version called SHA-256 (the SHA-2 family), where the number is represented as the hash length in bits.

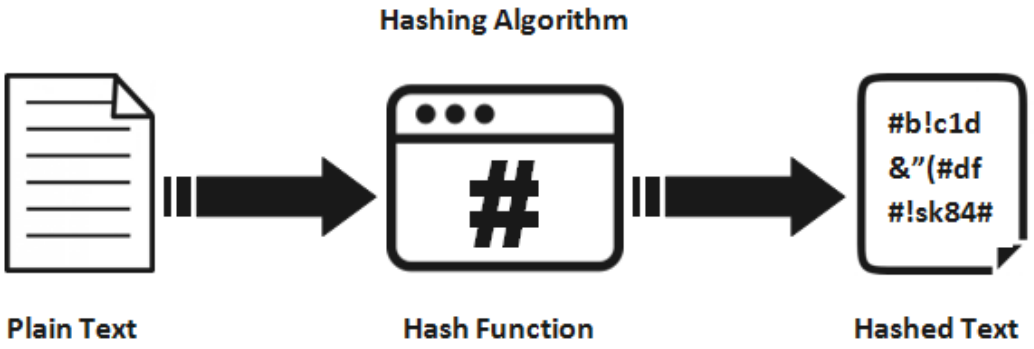


fig. 3.1.5 SHA Hashing Algorithm

Usage - This algorithm will be used in the Blockchain to check whether the model is sent by the correct clientID and Hashes it for Digital Signature to protect it from Leaks.

Steps - Preprocessing:
1. Padding - If we note M the message to be hashed, and l its length in bits where $l < 2^{64}$, then as a first step we create the padded message M', which is message M plus a right padding, such that M' is of length l', a multiple of 512.

2. Blocks - M' is parsed into N blocks of size 512 bits, M^1 to M^N, and each block is expressed as 16 input blocks of size 32 bits, M_0 to M_{15}.

3. Hash initialization. The initial hash value H^0 of length 256 bits (8 input blocks of 32 bits) is set by taking the first 32 bits of the fractional parts of the square roots of the first eight prime numbers:

$H_0^{(0)} = 6a09e667$
$H_1^{(0)} = bb67ae85$
$H_2^{(0)} = 3c6ef372$
$H_3^{(0)} = a54ff53a$
$H_4^{(0)} = 510e527f$
$H_5^{(0)} = 9b05688c$
$H_6^{(0)} = 1f83d9ab$
$H_7^{(0)} = 5be0cd19$

fig. 3.1.6 The eight input blocks of the initial has value H, in hexadecimal notation

Those values can be reproduced with the below snippet in most browsers and Node ≥8.2.1 (ECMAScript 2017):

```
for (i of [2, 3, 5, 7, 11, 13, 17, 19]) {
  input_block = parseInt((Math.sqrt(i) % 1).toString(2).slice(2, 34), 2);
  console.log(input_block.toString(16), input_block.toString(2).padStart(32, '0'));
}
```

fig. 3.1.7 Node snippet for producing hexadecimal values

3.2 Details of hardware and software

3.2.1 Hardware requirements

- OS: Microsoft Windows 8/10, Mac or Ubuntu 18 or higher
- CPU: Intel i5 7th generation or higher
- Memory: minimum 8GB of ram and 500GB Disk space

3.2.2 Software requirements

- Python: version 2.1 or above (recommended 3.3 and stable)
- Virtual Environment: To create an environment with all the dependencies
- Tmux: allows multiple terminal sessions to be accessed simultaneously in a single window.

3.3 Design Details

3.3.1 System Flow/ System Architecture

Components used:

- Clients: Users needing ML for any application(Segregation, Recommendation, Classification)
- Dataset - The users dataset on which ML needs to be applied
- Training Algorithm - Algorithm that trains the model with the Provided dataset
- Weights receiver - Receives the weights sent by client
- Weights aggregator - Aggregates the weights received from the client
- Blockchain - To secure our server from Byzantine attacks

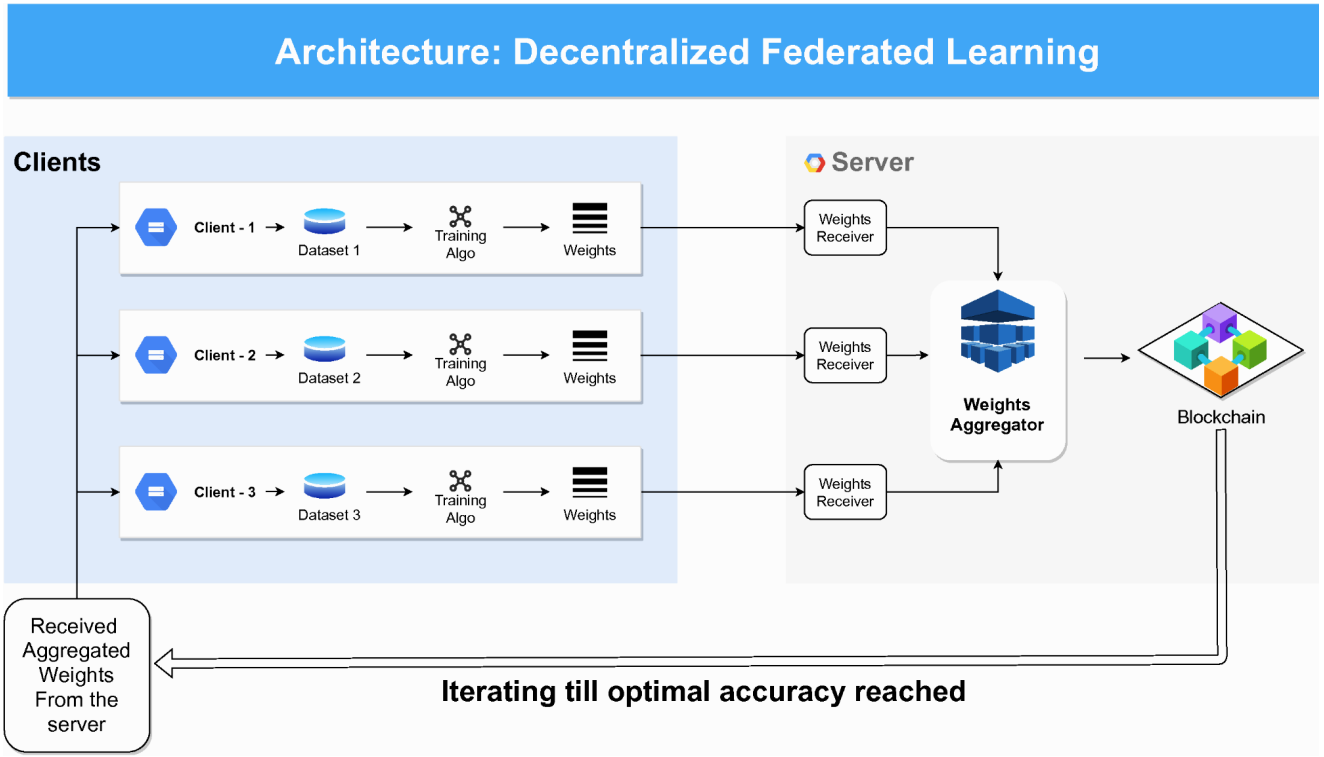


Fig. 3.3.1 System Architecture

3.3.2 Detailed Design

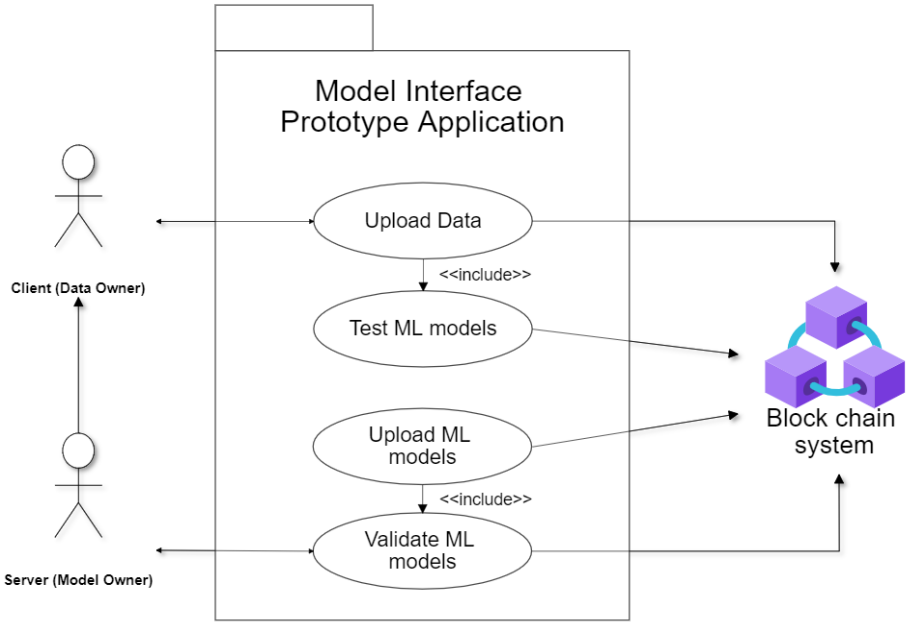


fig. 3.3.2.1 Detailed Design

In our model inference prototype, there are two model owners, in which the client can send data for evaluation to the test models which is used for new model validation and testing of already existing models. The server can upload the model, which once validated is committed to the blockchain system. Additionally, the Blockchain system participates in all use cases, as an external network, which is responsible for transaction recording. Either type of user acts as a participant during the data or model upload processes, and only after the majority of participants approve the transaction, it is committed to the blockchain.

In the model inference solution, federated learning participants can contribute their data and off-chain computed machine learning models for evaluation using the chain code, which is responsible for the data and model storage.

3.4 Methodology/Procedure (Approach to solve the problem)

Data that is stored on the client is fed to the training algorithm so that weights could be generated (using Linear regression for training the weights), After that all the local weights generated by the clients are received by weights receivers on the server. Then these weights from all weight receivers are aggregated with the help of a weight aggregator.

After this the aggregated weights are sent to the blockchain system for distribution where PoW(Proof of Work) and SHA256 (Secure Hash Algorithm) works and protects the System from Byzantine (Malicious user) attacks. This blockchain system distributes the aggregated weights back to the client, feeds the dataset and the received aggregated weights back to the training algorithm. This process repeats itself until the optimal accuracy is achieved.

CHAPTER 4
Results and Discussion
4.1 Results
The weights are passed from the client to the server for aggregation and then received back to the client.

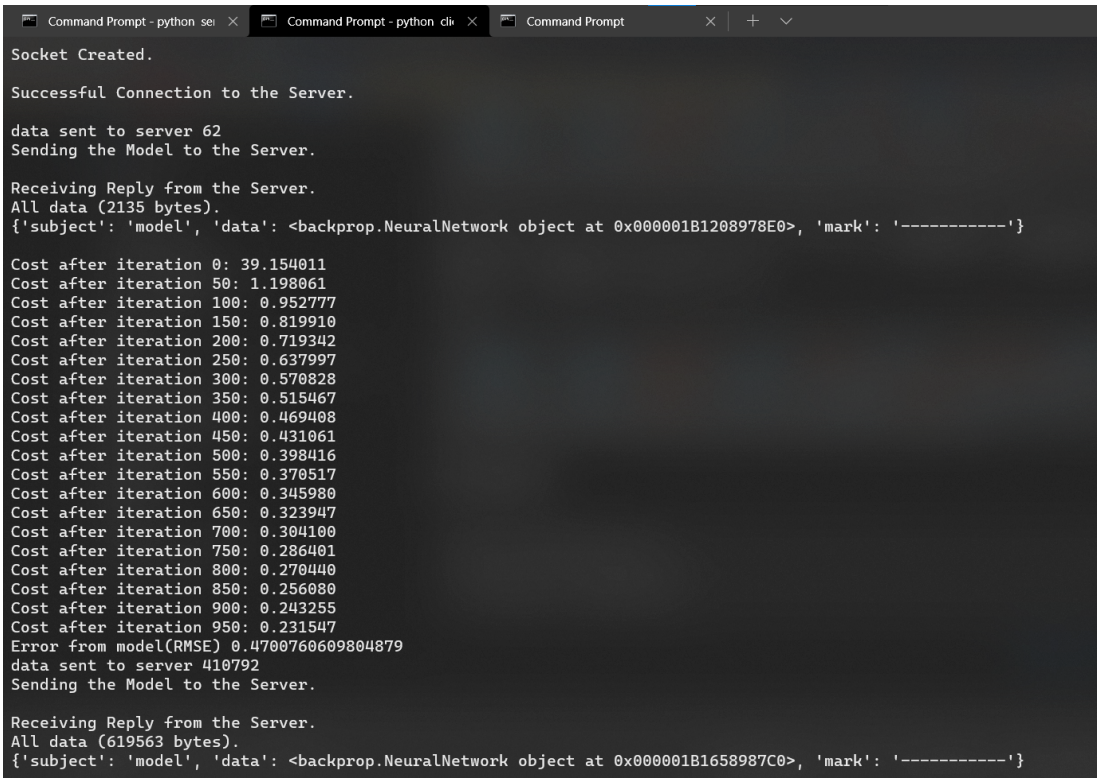


fig. 4.1.1 Output - Client 1

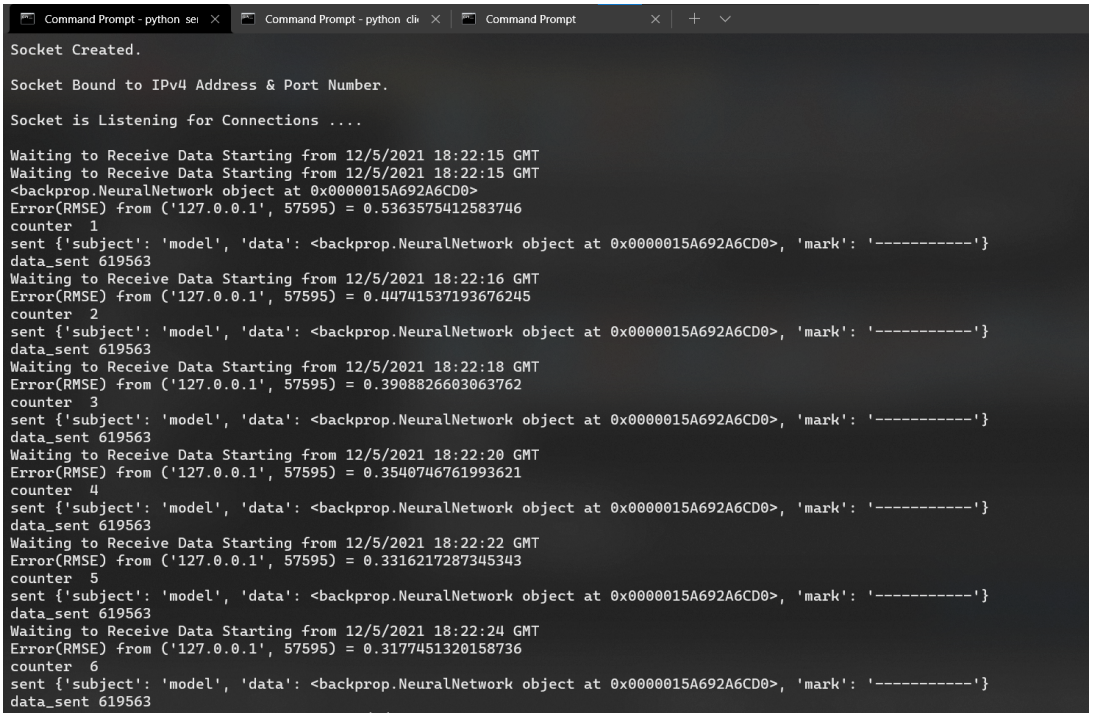


fig. 4.1.2 Output - Server

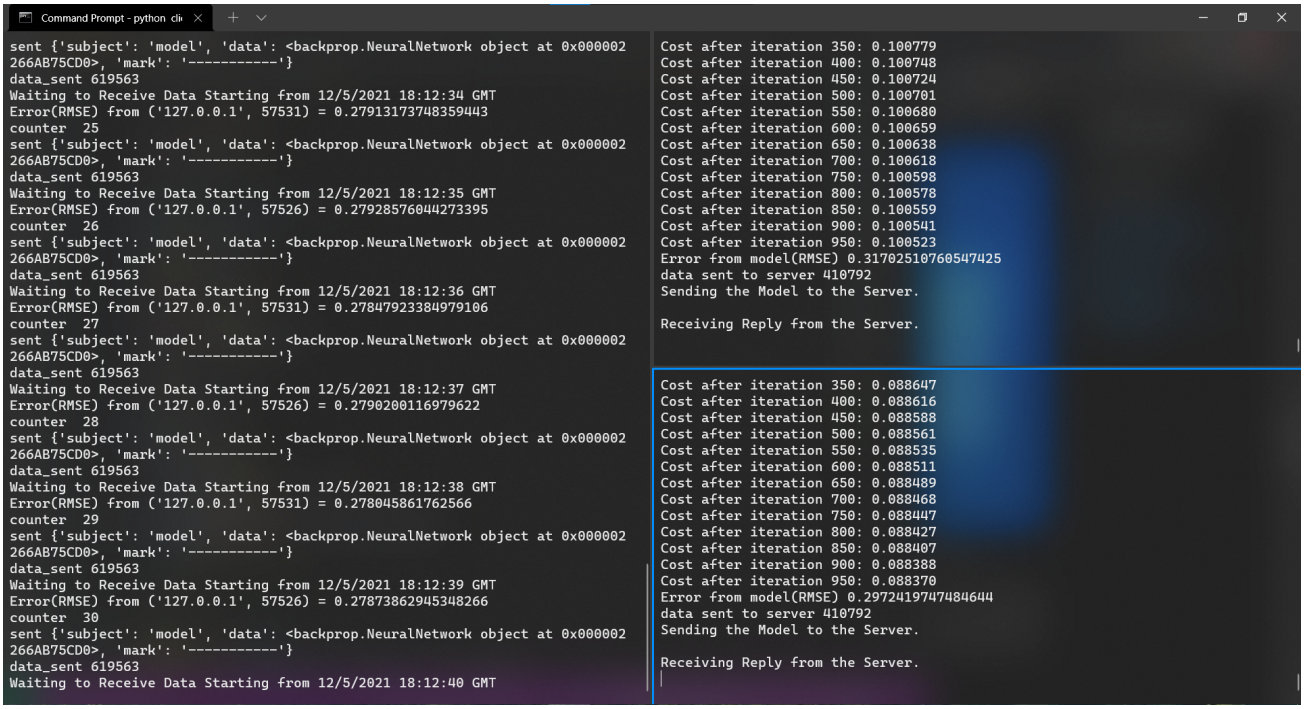
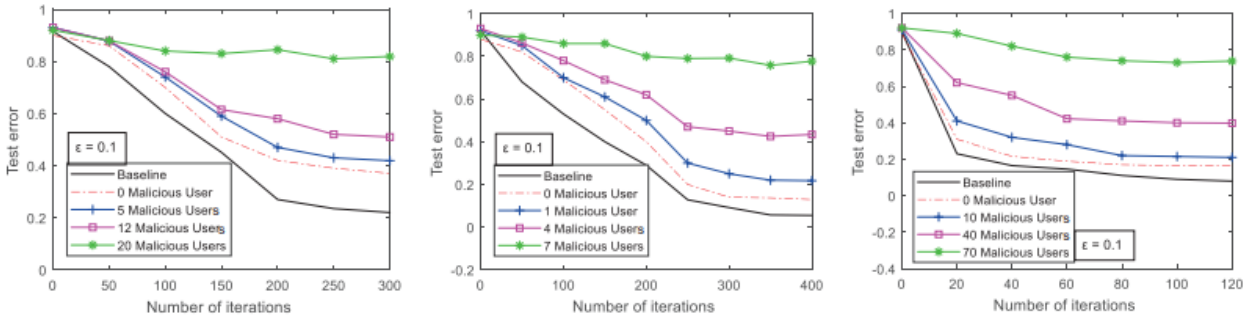


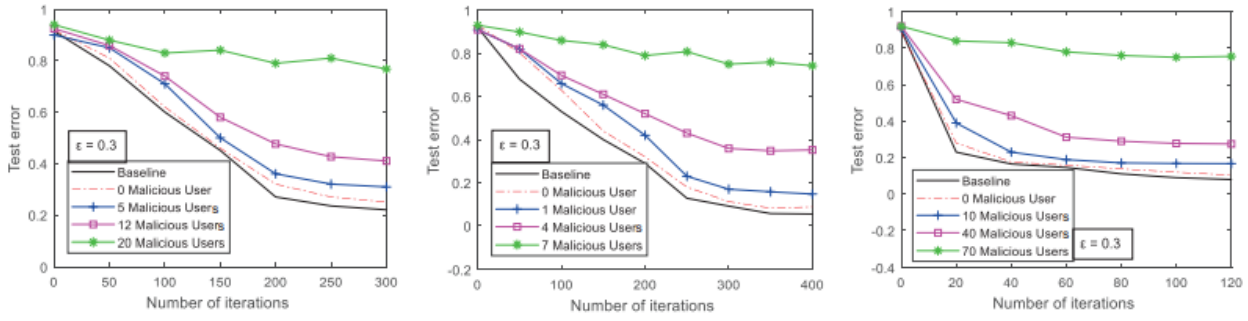
fig. 4.1.3 Output

4.2 Discussions

Provided are the graphs of test cases to prove that errors are reduced with the increase in iterations and for more number of Byzantine(Malicious user) attackers.



(a) Test error on synthetic data with $\epsilon = 0.1$ (b) Test error on breast cancer data with $\epsilon = 0.1$ (c) Test error on MNIST data with $\epsilon = 0.1$



(d) Test error on synthetic data with $\epsilon = 0.3$ (e) Test error on breast cancer data with $\epsilon = 0.3$ (f) Test error on MNIST data with $\epsilon = 0.3$

fig 4.2.1 Discussion - graph

CHAPTER 5

Conclusion and Future Work

In this project, we implemented a privacy-preserving federated learning scheme using Blockchain and Machine Learning. We also discussed the privacy related issues and threats to the model in a general federated learning approach and tried to use blockchain to deal with malicious clients and data.

We have looked at both the implementation of federated learning with, and without blockchain in which combination federated learning and blockchain. But we need to improve the security and performance of the system by the following ways:

1) Reach to **Serverless** architecture by remove model owner and shifting the secure aggregation to the Blockchain system itself
2) Design a System with a mixed approach of **Blockchain and Secret Sharing**.
3) **Scaling** and designing of the distributed systems for a large number of clients.
4) Using Blockchain framework like **Hyperledger Fabric** to develop a serverless private blockchain system

REFERENCES

[1] Akihito Taya, Takayuki Nishio, Masahiro Morikura,Koji Yamamoto " Decentralized and Model-Free Federated Learning: Consensus-Based Distillation in Function Space| Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, January 2020.

[2] Runhua Xu , Nathalie Baracaldo , Yi Zhou , Ali Anwar , James Joshi , Heiko Ludwig , "*FedV: Privacy-Preserving Federated Learning over Vertically Partitioned Data",* IEEE, March 2021

[3] Dinh C. Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N. Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, H. Vincent Poor, "*Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges"* : IEEE, April 2021

[4] https://ai.googleblog.com/2017/04/federated-learning-collaborative.html