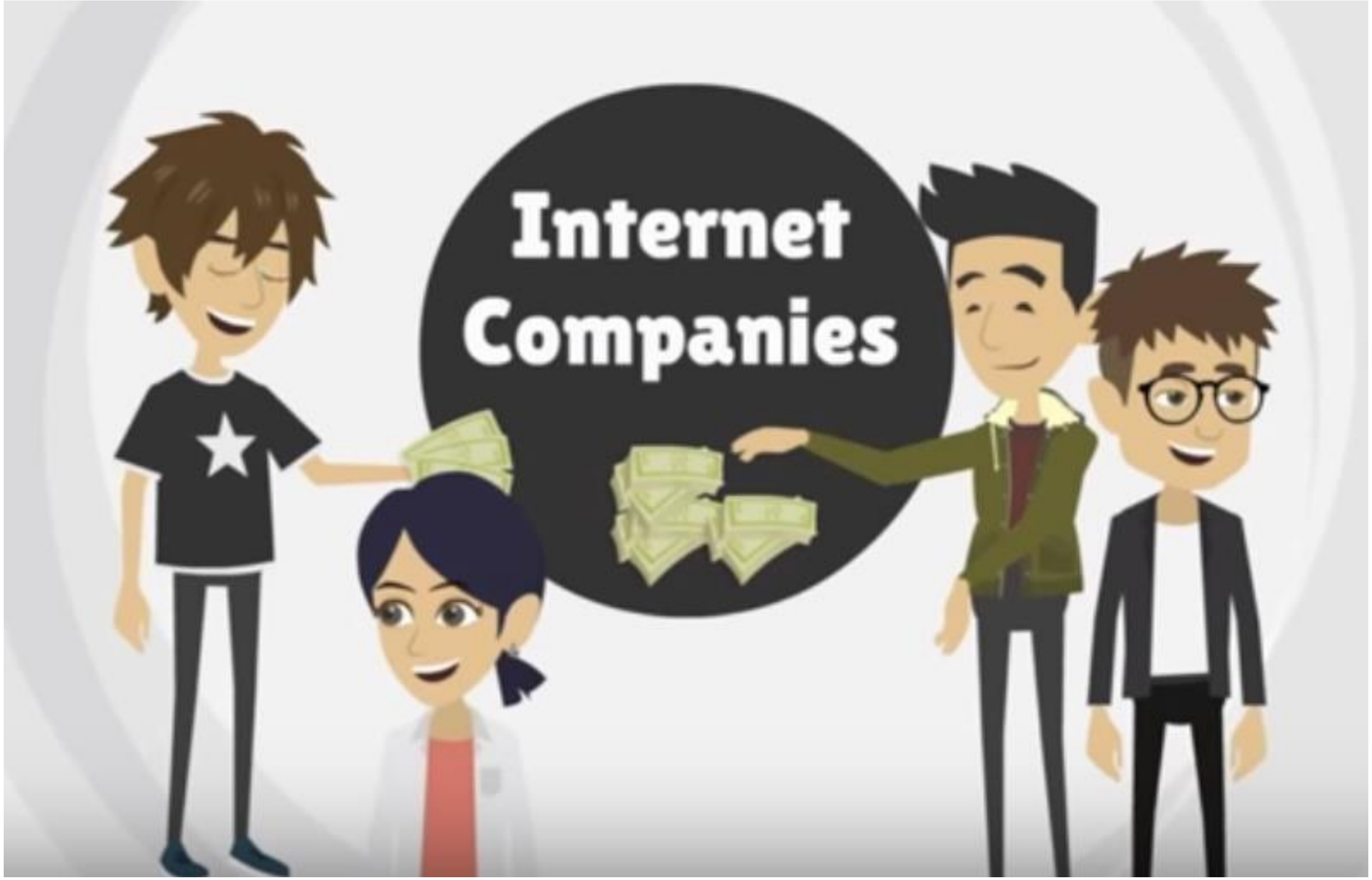
Blockchain for federated learning

Sreya Francis, Martinez Ismael

Current Scenario



Existing technology - Issues

- Data collection means adopted right now is incredibly privacy invasive
- We give our data for free in return of a free service
- Latency issues
- High transfer costs
- Centralized ownership (Users don't participate in the current system)
- Very limited data for healthcare research

Current Issues

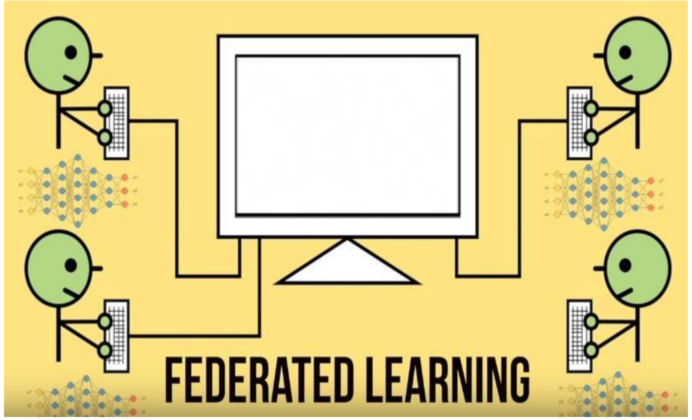
- Privacy Concerns
  - We don’t have control over the data we generate!
- We are losing one source of natural income
  - Data is our natural resource and we own it
- Sensitive Product Problem - some services are creepy
  - High risks of theft, embarrassment, resale ……etc
- Centralized control by Big Tech Giants
  - All of our data are controlled by tech giants like google, facebook

How can we solve this?

- Enhance user privacy
  - We should control our data
- We should be rewarded for the data we own
  - Rewards based on data quality and quantity
- Decentralized power
  - Everyone has control over their data
- Enhance production of sensitive products/models
  - Enhanced privacy would make it easier to collect data related to sensitive fields like healthcare

Ingredients for the solution

Federated Learning



BlockChain

Internet of Things

Cryptography

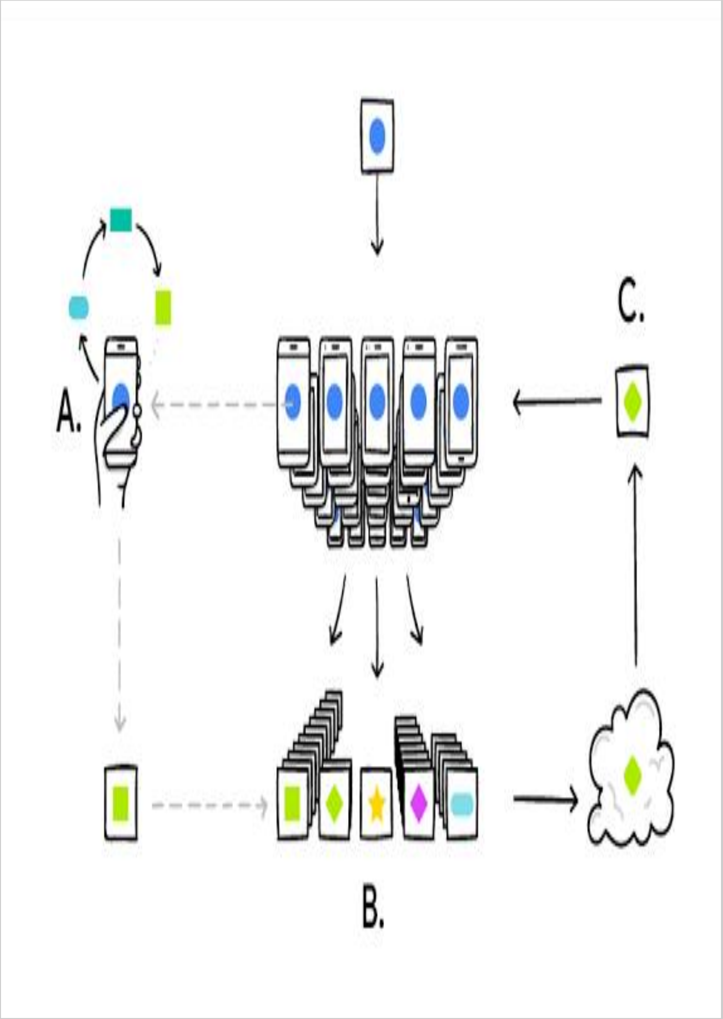
1

Federated Learning

- What is Federated Learning?
- How does it work?
- Federated Learning Platforms

Federated Learning - Definition

- **Idea:** machine learning over a distributed dataset
- **Federated computation**: where a server coordinates a fleet of participating devices to compute aggregations of devices' private data.
- **Federated learning:** where a shared global model is trained via federated computation.
- **Definition:** training a shared global model, from a federation of participating devices which maintain control of their own data, with the facilitation of a central server.



Federated Learning – Brief stepwise overview

- **Step 1:** Users download a **Model**
- **Step 2:** Users train the **Model** on their own data.
- **Step 3:** Users upload their **Gradients** to a server
- **Step 4:** **Gradients** are added up to protect privacy.
- **Step 5:** The **Model** is updated with the **Global Model**.

Federated Learning – Algorithm

Server

Until Converged:

1. Select a random subset (e.g.200) of the (online) clients

2. In parallel, send current parameters **θ(t)** *to those clients*

Selected client K

1.Receive **θ(t)** from server.

2. Run some number of minibatch SGD steps, producing **θ'**

3. Return **θ'-θ(t)** to server.

3. **θ(t+1)** *=* **θ(t)** *+ data-weighted average of client updates*

Federated Learning – Pros & Cons

Pros:

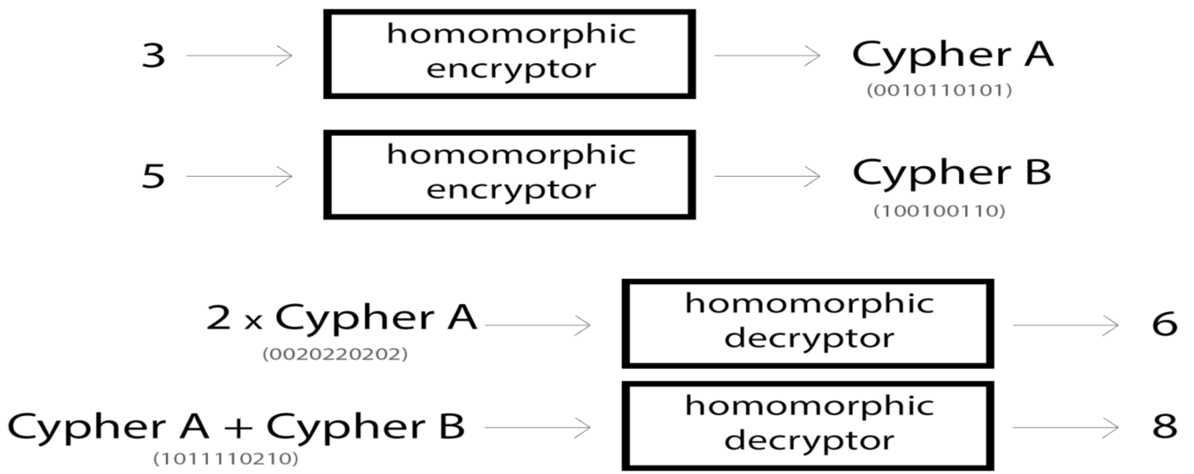
- **Enhanced User Privacy:** Users keep their data secret

Cons:

- **Privacy:** Gradients give hints about data
- **Theft:** Participants can steal the updated models
- **No Sensitive Products:** Because of theft/privacy issues

One Possible Solution: Homomorphic Encryption

What is Homomorphic Encryption?



- Homomorphically encrypt the user **gradients** so that the gradient privacy is preserved
- Privacy-Preserving **Deep Neural Network model** (2P-DNN) based on the **Paillier** Homomorphic **Cryptosystem** could be used to enhanced global model privacy
- Hence there is no issue of theft or privacy intrusion in this case

Reward Calculation

Possible way

- Based on user model performance on validation set
  - To evaluate the validity of user data, we can run a validation check on the user model based on a trusted validation set.
  - Based on the performance on validation set, the users can be rewarded.
  - If the validation accuracy goes below a specified threshold, the data is rejected.

- Pros
  - An easy and fast way to calculate user reward immediately after client side training

- Cons
  - At any given iteration, an honest gradient may update the model in an incorrect direction, resulting in a drop in validation accuracy.
  - This is confounded by the problem that clients may have data that is not accurately modeled by our trusted validation set

Issues with data in FL

What can go wrong?

- Gamber attack
  - User/Attacker can randomly pick data and maliciously change them
  - User can give garbage input
  - User/Attacker give data that does not contribute to the model

- Omniscient attack
  - Attackers are supposed to know the gradients sent by all the workers
  - Use the sum of all the gradients, scaled by a large negative value,
  - And replace some of the gradient vectors.

- Gaussian attack
  - Some of the gradient vectors are replaced by random vectors sampled from a Gaussian distribution with large variances.

How to counter adversaries?

Possible ways

- Based on KRUM Algorithm
  - Uses the Euclidean distance to rank the gradients
  - Determines which gradient contributions are removed
  - the top f contributions to the client model that are furthest from the mean client contribution are removed from the aggregated gradient

- Pros
  - specifically designed to counter adversaries in federated learning.

- Cons
  - Not an absolute measure of user contribution
  - Implementation is a bit complicated

How to ensure validity of gradients?

Possible ways

Let us assume that q out of n vectors are Byzantine/incorrect, where q < n:

Krum's Algo in a nutshell:

$$Krum(\{\tilde{v}_i : i \in [n]\}) = \tilde{v}_k,$$
$$k = \underset{i \in [n]}{\operatorname{argmin}} \sum_{i \rightarrow j} \|\tilde{v}_i - \tilde{v}_j\|^2,$$

where $i \rightarrow j$ is the indices of the $n-q-2$ nearest neighbours of $\tilde{v}_i$ in $\{\tilde{v}_i : i \in [n]\}$ measured by Euclidean distance.

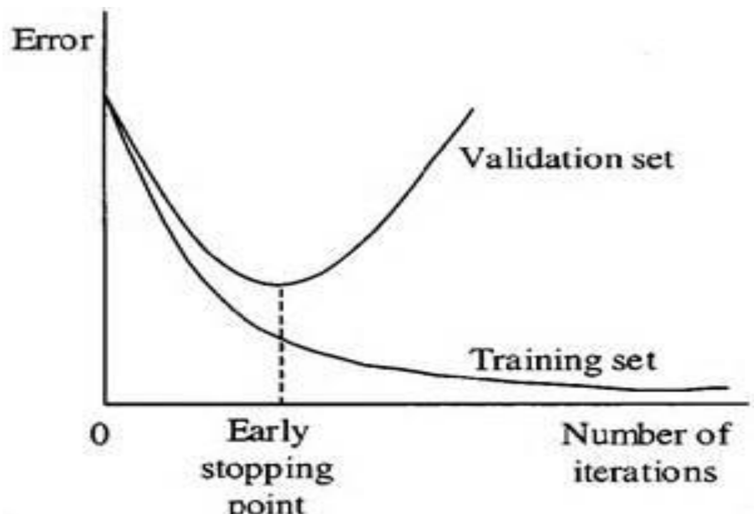
- Works only when q < n
- Ensure upto 33% protection against adversarial attacks
- Best solution proposed till date

Proposed Solution to the User Reward Issue

- Data Cost
  - Each User calculates his/her data cost
  - Class id – Ci, Number of samples - Nci
  - Cost per user -> ∑j=1 to k (j*Nci)

- Generate validation set
  - Based on parameters passed to calculate data cost
  - Automatically generate a validation set with some random samples
  - Samples pertain to user specified classes

- Training



•Stop training before the model over-fits data
•If validation error doesn't go down, user entry is wrong
•If validation error goes down, user entry is valid and pay the user based on calculated data cost

2

To Do: Causal Learning

- How can Causal Learning help FL?
- Issues?
- Possible solutions