Federated Machine Learning With Python

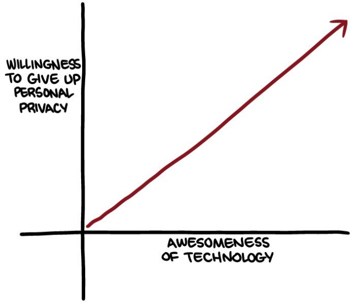
Training models without looking at data

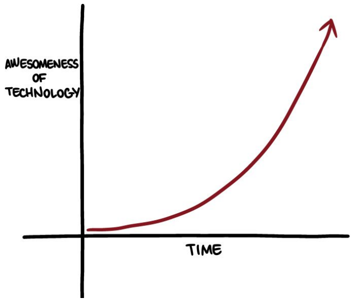
Dhanshree Arora
MLOps Engineer
EDER LABS

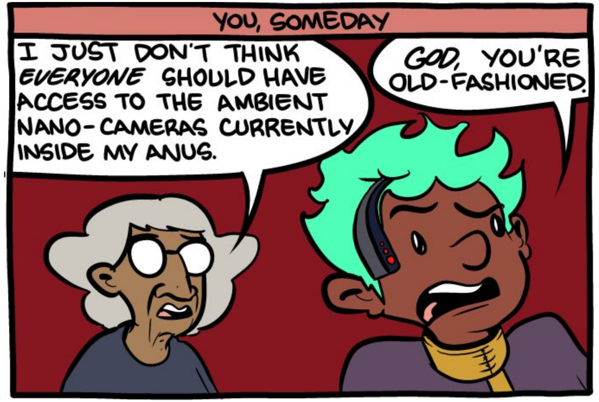
Agenda

1. The Privacy Cost of Machine Learning
2. Privacy Preserving Machine Learning
3. Federated Learning
4. Building a Minimal FL System
5. Opportunities in FL
6. Conclusion
7. Questions

The Privacy Cost in Machine Learning

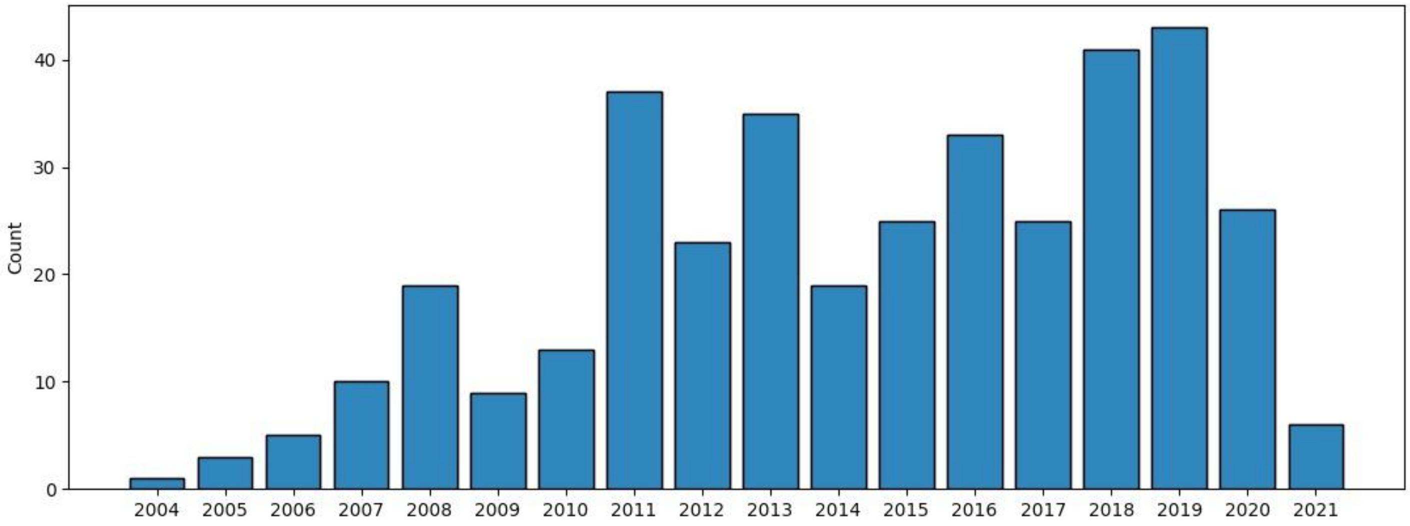






Credits: [SMBC]

Data breaches through the ages

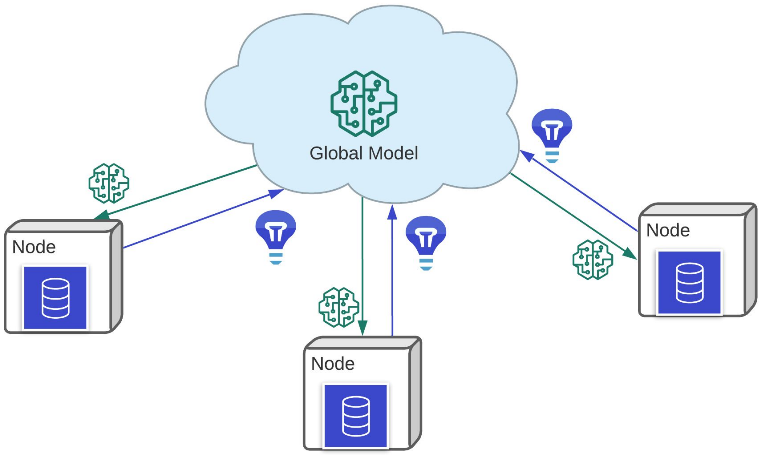


Data credit: [Information is Beautiful]

Privacy Preserving Machine Learning (PPML)

Federated Learning

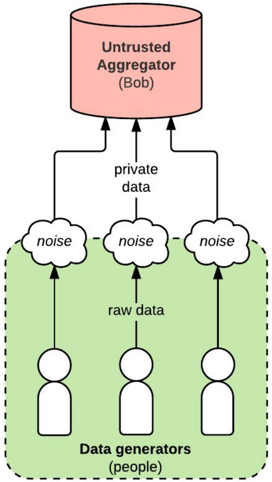
Training models on data at its source



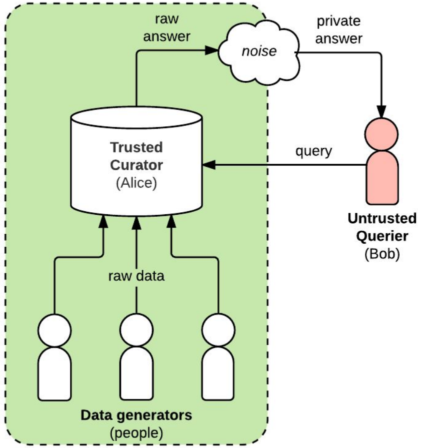
Differential Privacy

Adding noise to de-identify data while preserving the distribution and relationships within the data

q(data + noise)



Local privacy



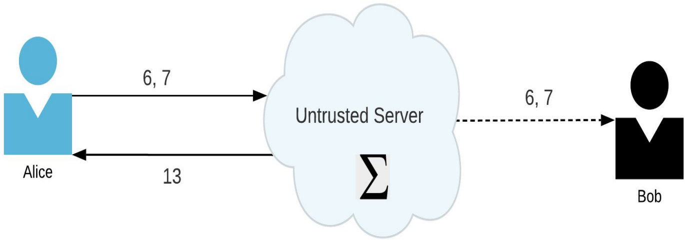
Global privacy

q(data) + noise

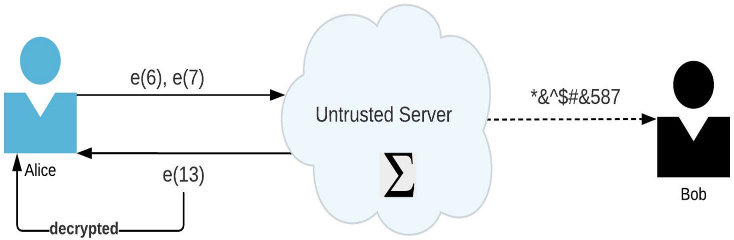
Credits: [Quantalabs/Differential-Privacy]

Homomorphic Encryption

Mathematical operations on encrypted data



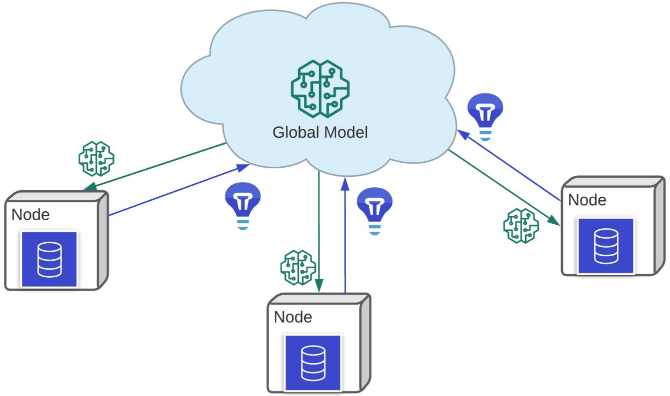
Without HE

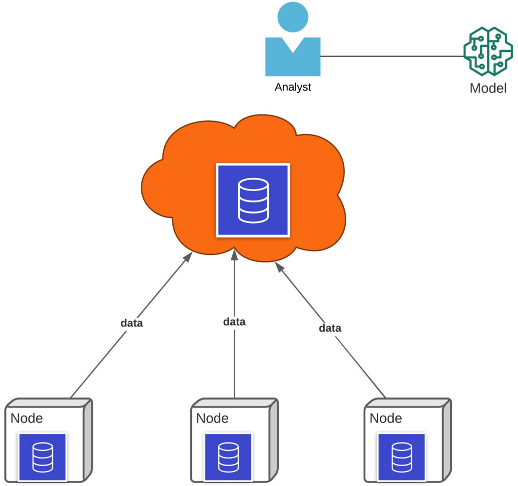


With HE

Federated Learning

Federated Learning vs Centralized Learning





Federated Learning Use-Cases

EDGE DEVICES:

- Recommendation
- Routine device storage maintenance
- Health Monitoring
- Predictive Typing
- Facial Unlocking

ENTERPRISE:

- Credit Card Fraud Detection
- Credit Lending
- Disease Prediction
- Sentiment analysis
- Autonomous vehicles
- Precision Medicine

Horizontal Federated Learning

sex	age	spo2	comorbidities	symptoms	oxygen_req	icu_num_days
m	52	74	0	...	200	7
f	23	89	1	...	150	13
f	42	90	0	...	190	5

Hospital A

sex	age	spo2	comorbidities	symptoms	oxygen_req	icu_num_days
m	25	70	0	...	120	19
f	32	85	1	...	120	4
m	68	65	0	...	210	11

Hospital B

sex	age	spo2	comorbidities	symptoms	oxygen_req	icu_num_days
m	46	74	0	...	150	7
m	84	89	1	...	300	20
f	39	90	0	...	200	10

Hospital C

Vertical Federated Learning

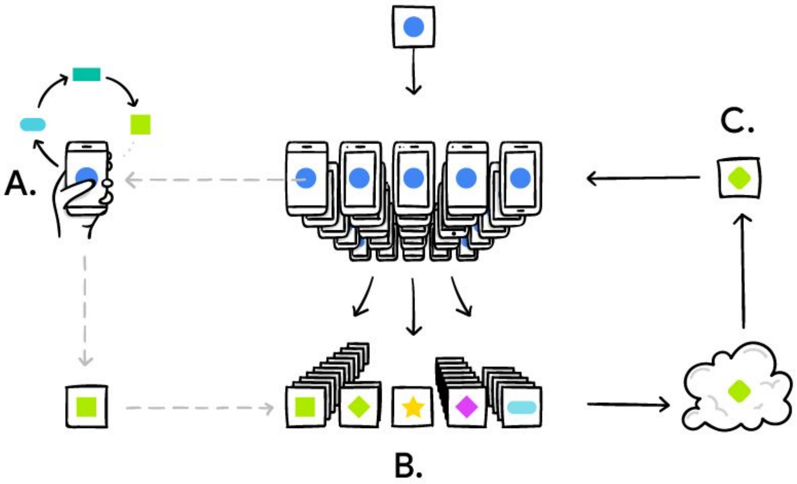
Hospital A

name	sex	age	spo2	comorbidities	symptoms	oxygen_req	icu_num_days
Person A	m	52	74	0	...	200	7
Person B	f	23	89	1	...	150	13
Person C	f	42	90	0	...	190	5

Fitness Tracking App

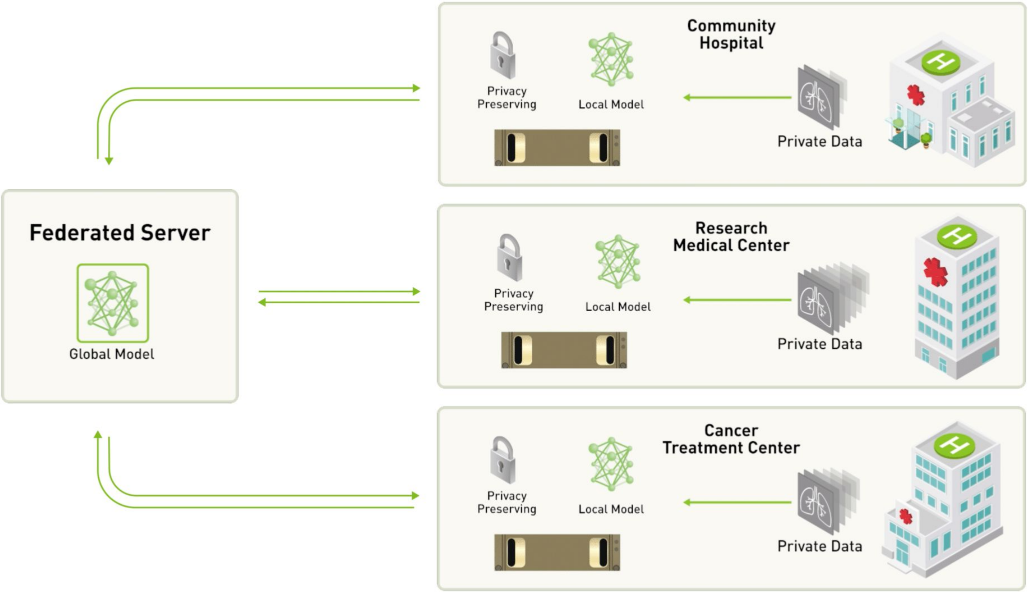
name	avg_active_mins	avg_rhr	avg_sleep	avg_emot
Person A	120	65	8h30m	happy
Person B	40	70	5h30m	uninterested
Person S	300	52	6h30m	uninterested

Cross Device Federated Learning



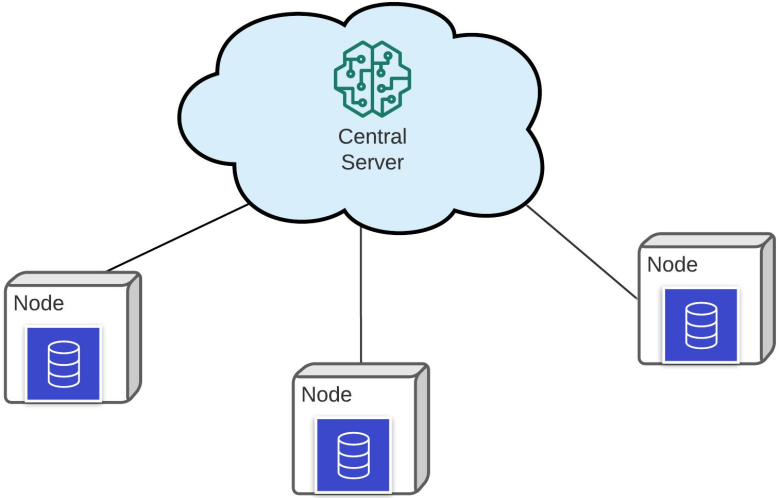
Credits: [Google AI Blog - Collaborative Machine Learning without Centralizing Data]

Cross Silo Federated Learning

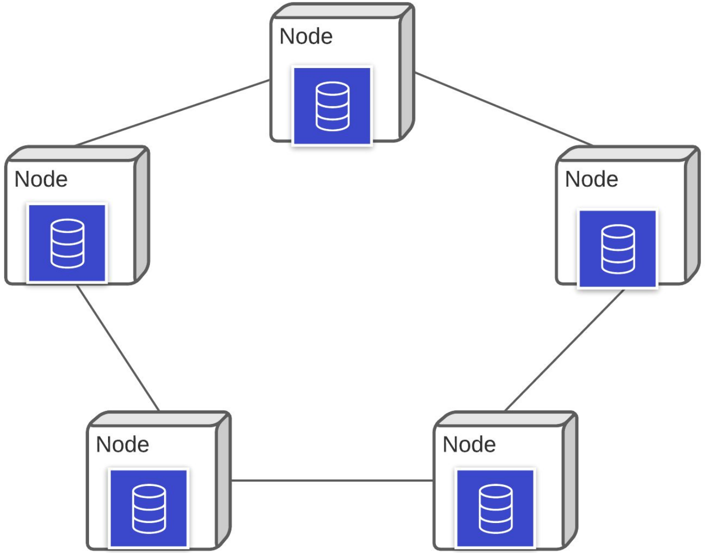


Credits: [Nvidia Blog - What is Federated Learning]

Centralized Federated Learning



Decentralized Federated Learning



Federated Aggregation

- Area of active research
- Simple approach: aggregate weights or gradients from local models.
- Secure aggregation: aggregate encrypted local updates and decrypt the result.
- Several caveats, discussed in the Challenges section.

Building a Minimal FL System

The Ingredients

- Centrally Coordinating Server
- A modelling and data processing utility
- A communication channel - we use websockets
- A medium to transfer local updates - we use Kafka
- Naive model averaging
- Tracking History

Base Example: This tutorial

A Small Note

- Socketio - enables real-time bidirectional event-based communication between clients and a server.
- Kafka - a distributed event or message streaming platform that allows you to work with a Producer Consumer pattern.

The Recipe - Server

```python
class Server:
    def __init__(self):
        pass

    async def connect(self, sid, environ):
        pass
```

```python
class Server:
    def __init__(self):
        pass

    async def connect(self, sid, environ):
        # connect with nodes, start training on min nodes

    async def start_round(self):
        # start a training round and send global model
```

```python
class Server:
    def __init__(self):
        pass

    async def connect(self, sid, environ):
        # connect with nodes, start training on min nodes

    async def start_round(self):
        # start a training round and send global model

    async def fl_update(self, sid, data):
        # receive ack for updates

    def consume_updates(self):
        # consume updates when all updates are received
```

```python
class Server:
....
    async def fl_update(self, sid, data):
        # receive ack for updates

    def consume_updates(self):
        # consume updates when all updates are received

    def aggregate(self, client_mapped_weights):
        # aggregate weights for layers with trainable weights

    def evaluate(self, aggregated_weights):
        # Evaluate on a holdout set

    def store_history(self):
        # Store federated losses across rounds
```

The Recipe - Client

```python
class Node:
    def __init__(self):
        pass

    async def connect(self, sid, environ):
        # Connect to the server
```

```python
class Node:
    def __init__(self, address, partition, client, epochs):
        pass

    def connect(self):
        # Connect to server

    def connection_received(self):
        # Get ack from server
```

```python
class Node:

    def start_training(self, _model):
        # get model from json
        # compile model
        # fit
        # evaluate
        # send updates
        pass

    def fit(self, model):
        pass

    def send_updates(self, loss):
        # encode individual layers as b64 strings
        # produce over the updates topic on a Kafka server
        pass
```

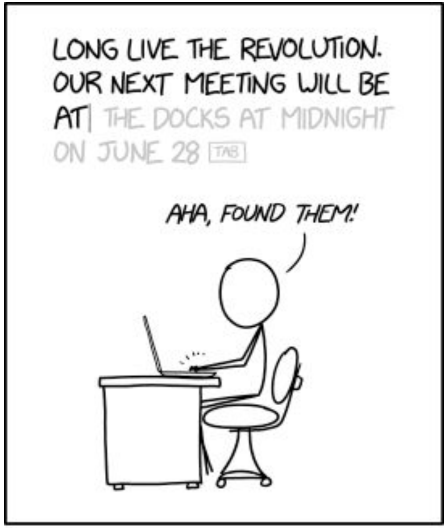
```python
class Node:


    def end_session(self, data):
        # get latest model weights
        # update local model
        # Clean up as necessary
        pass

    def disconnect(self):
        # Disconnect signal from server
```

Opportunities in FL

- The Non IID data conundrum
- Collaborations and Partnerships are difficult
  - No control over data collection stack
  - How do you do EDA?
- Technical failures:
  - Network latency
  - Connection dropouts
  - Corrupted local updates

- Not a standalone solution to privacy:
    - Add noise by way of Differential Privacy

- Larger attack surface with multiple nodes, if not implemented correctly.



WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

Conclusion

Build Privacy into solutions proactively.

Thanks!

Find me here:

Additional Resources:

- Code
- Advances and Open Problems in FL
- Google Federated Web comic