University of Idaho
Department of Computer Science

# CS 404/504 Special Topics: Adversarial Machine Learning

*Dr. Alex Vakanski*

# Lecture 13

## Defenses against Privacy Attacks

# Lecture Outline

- Defenses against Privacy Attacks
  - Anonymization techniques
  - Encryption techniques
  - Differential privacy
  - Distributed learning
  - ML-specific techniques
- Jason Starace presentation
  - Introduction to differential privacy
- Lu Cai presentation
  - Differentially private SGD
- Johnny Stuto presentation
  - Scalable private learning with PATE
- Sohag Sharidur
  - Introduction to federated learning

# Defenses against Privacy Attacks

*Defenses against Privacy Attacks*

- ***Data privacy*** techniques have the goal of allowing analysts to learn about *trends* in data, without revealing information specific to *individual data instances*
  - Therefore, privacy techniques involve an intentional release of information, and attempt to control what can be learned from the released information
- Related to data privacy is the ***Fundamental Law of Information Recovery***, which states that "*overly accurate estimates of too many statistics can completely destroy privacy*"
  - I.e., extracting useful information from a dataset (e.g., for training an ML model) poses a privacy risk to the data
- There is an inevitable trade-off between privacy and accuracy (i.e., utility)
  - Preferred privacy techniques should provide an estimate of how much privacy is lost by interacting with data

# Defenses against Privacy Attacks

*Defenses against Privacy Attacks*

- Defense strategies against privacy attacks in ML can be broadly classified into:
    - Anonymization techniques
    - Encryption techniques
    - Differential privacy
    - Distributed learning
    - ML-specific techniques

# Anonymization Techniques

- ***Anonymization*** techniques provide privacy protection by removing identifying information in the data
- E.g., remove personal identifiable information (PII)
  - In the example below, the Name and Address columns are removed

| User ID | Name | Address | Account Type | Subscription Date |
|---|---|---|---|---|
| 001 | Alice | 123 A St | Pro | 01/02/20 |
| 002 | Bob | 234 B St | Free | 02/03/21 |
| 003 | Charlie | 456 C St | Pro | 03/04/18 |

# Anonymization Techniques

*Anonymization Techniques*

- Anonymization is not an efficient defense method, since the remaining information in the data can be used for identifying the individual data instances
  - For example, based on health records (including diagnoses and prescriptions) with removed personal information released by an insurance group in 1997, a researcher extracted the information for the Governor of Massachusetts
    - This is referred to as ***de-anonymization***
  - The same researcher later showed that 87% of all Americans can be uniquely identified using 3 bits of information: ZIP code, birth date, and gender

Dataset 1: Users medical database

| User ID | Name | Address | Zip Code | Birth date | Gender | Probable disease ID |
|---|---|---|---|---|---|---|
| 001 | Alice | 123 A St | 83401 | 01/02/1997 | F | 120 |
| 002 | Bob | 234 B St | 83402 | 02/03/1995 | M | 35 |
| 003 | Charlie | 456 C St | 83403 | 03/04/1999 | M | 240 |

Dataset 2: Users medical database with name and address removed

| User ID | Zip Code | Birth date | Gender | Probable disease ID |
|---|---|---|---|---|
| 001 | 83401 | 01/02/1997 | F | 120 |
| 002 | 83402 | 02/03/1995 | M | 35 |
| 003 | 83403 | 03/04/1999 | M | 240 |

# Linkage Attack

*Anonymization Techniques*

- De-anonymization of data by using connections to external sources of information is referred to as *linkage attack*
  - For example:
    - In 2006, Netflix published anonymized 10 million movie rankings by 500,000 customers
    - Two researchers showed later that by using movie recommendations on IMDb (Internet Movie Database) they could identify the customers in the Netflix data

Dataset 1: Annonymized dataset with removed personal information

| User ID | Name | Address | Account Type | Subscription Date |
|---|---|---|---|---|
| 001 |  |  | Pro | 01/15/20 |
| 002 |  |  | Pro | 02/03/21 |
| 003 |  |  | Free | 03/04/18 |

Dataset 2: External public dataset that reveals the users in Dataset 1

| User ID | Product Name | Product Price | Purchase Date |
|---|---|---|---|
| 001 | TV | 400 | 01/02/20 |
| 002 | Iphone | 1,199 | 02/02/21 |
| 003 | Watch | 130 | 02/22/18 |

# *k*-anonymity

*Anonymization Techniques*

- ***k-anonymity*** is an approach for protecting data privacy by suppressing certain identifying data features
  - This approach removes fields of data for individuals who have unique characteristics
    - E.g., students at UI who are from Latvia and are enrolled in Architecture
- A dataset is *k*-anonymous if for any person's record, there are at least $k-1$ other records that are indistinguishable
- Limitation: this approach is mostly applicable to large datasets with low-dimensional input features
  - The more input features there are for each record, the higher the possibility of unique records

# Encryption Techniques

*Encryption Techniques*

- ***Encryption*** is a cryptography approach, which converts the original representation of information into an alternative form
  - The sender of encrypted information shares the decoding technique only with the intended recipients of the information
- Encrypting the training data has been applied in ML
  - Common techniques for data encryption include:
    - Homomorphic encryption (HE)
    - Secure multi-party computation (SMPC)
- Encrypting ML models is less common approach
  - Homomorphic encryption has been applied to the model gradients in collaborative DL setting to protect the model privacy

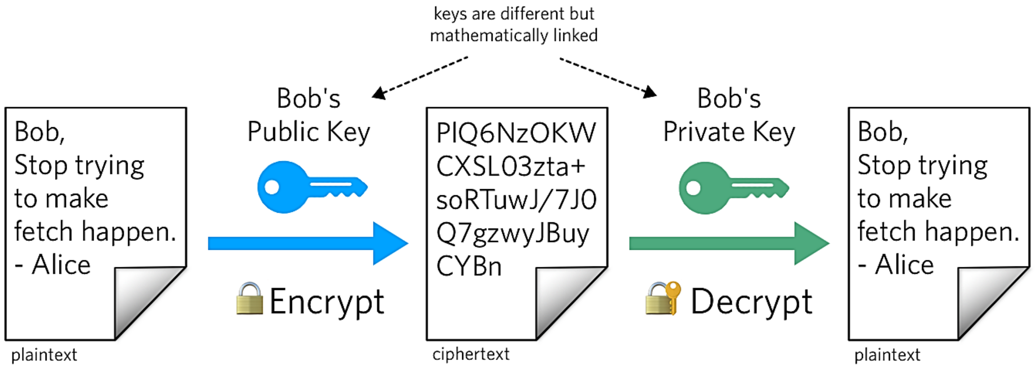

keys are different but mathematically linked
Bob's Public Key
Bob's Private Key
Bob, Stop trying to make fetch happen. - Alice
plaintext
Encrypt
PIQ6NzOKW CXSL03zta+ soRTuwJ/7J0 Q7gzwyJBuy CYBn
ciphertext
Decrypt
Bob, Stop trying to make fetch happen. - Alice
plaintext

Figure form: What is Public Key Cryptography?

# Homomorphic Encryption

*Encryption Techniques*

- ***Homomorphic encryption (HE)*** allows users to perform computations on encrypted data (without decrypting it)
  - Encrypted data can be analyzed and manipulated without revealing the original data
- HE uses a public key to encrypt the data, and applies an algebraic system (e.g., additions and multiplications) to allow computations while the data is still encrypted
  - Only the person who has a matching private key can access the decrypted results

# Homomorphic Encryption

*Encryption Techniques*

- In ML, training data can be encrypted and send to a server for model training
  - Even if the server is untrusted or it is compromised, confidentiality of the data will remain preserved
  - One main limitation of HE is the slowing down of the training process
- HE has been applied to traditional ML approaches, such as Naïve Bayes, Decision Trees
  - Training DNNs over encrypted data is still challenging, due to the increased computational complexity

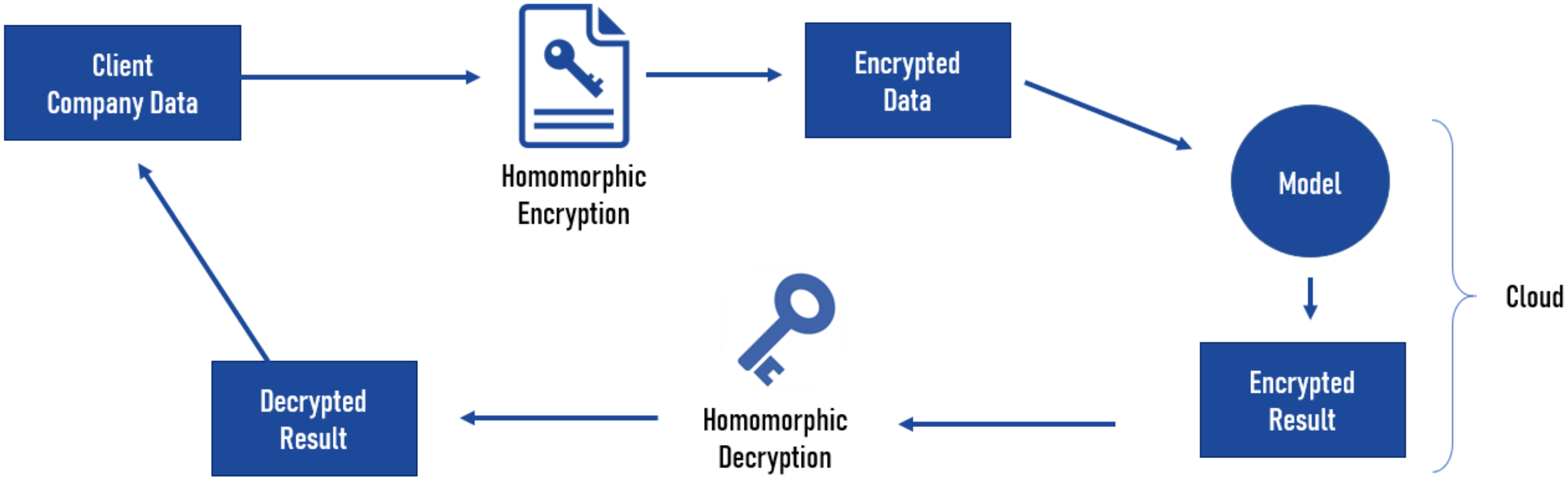


Figure form: Homomorphic Encryption & Machine Learning: New Business Models

# Privacy versus Confidentiality

*Encryption Techniques*

- Encryption techniques in ML are mainly applied to protect the confidentiality of the data or model
- *Confidentiality* refers to keeping the information (training data, model parameters) hidden from the clients and the public
    - It is ensuring that only authorized parties have access to the information
    - E.g., a server has an ML model trained on private data and provides the model to a client for inference
        - It is preferred to preserve the confidentiality of the model parameters from the client
- *Privacy* refers to intentional release of information in a controlled manner to prevent unintended information leakage
    - It is ensuring that released data cannot uniquely identify individual inputs
    - E.g., a server applies Differential Privacy to a trained ML model to prevent memorization of information about individual inputs
- Protecting privacy is more challenging than protecting confidentiality

# Secure Multi-Party Computation

*Encryption Techniques*

- ***Secure Multi-Party Computation*** (SMPC) is an extension of encryption in multi-party setting
  - SMPC allows two or more parties to jointly perform computation over their private data, without sharing the data
  - E.g., two banks want to know if they have both flagged the same individuals and learn about the activities by those individuals
    - The banks can share encrypted tables of flagged individuals, and they can decrypt only the matched records, but not the information for individuals that are not in both tables

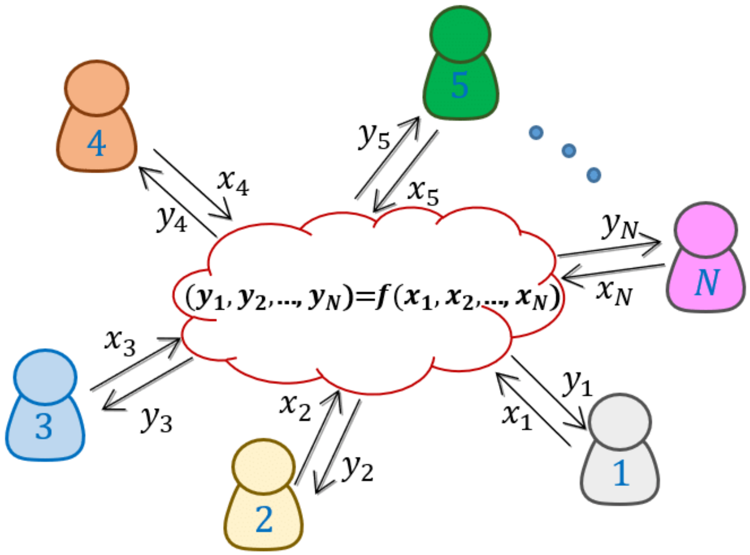


Figure form: Generation and Distribution of Quantum Oblivious Keys for Secure Multiparty Computation

# Secure Multi-Party Computation

*Encryption Techniques*

- SMPC versus HE
  - SMPC protects the privacy of the data in collaborative learning
    - E.g., participants in collaborative learning do not trust the other participants or the central server
  - HE protects the confidentiality of the data from external adversaries
    - E.g., a data owner wants to use a MLaaS (Machine Learning as a Service), but does not trust the service provider: (1) the owner sends encrypted data, (2) the provider processes encrypted data and sends back encrypted results, (3) the owner decrypts the results
    - Or, a bank can store encrypted banking information in the cloud, and use HE to ensure that only the employees of the bank can access the data

# Secure Multi-Party Computation

*Encryption Techniques*

- In ML, SMPC can be used to compute updates of the model parameters by multiple parties that have access to their private data
  - For examples, SMPC has been applied to federated learning, where participants encrypt their updates, and the central server can recover only the sum of the updates from all participants
  - Beside the data privacy, SMPC also offers protection against adversarial participants
    - Either all parties are honest and can jointly compute the correct output, or if a malicious party is dishonest the joint output will be incorrect
- SMPC has been applied to traditional ML models, such as decision trees, linear regression, logistic regression, Naïve Bayes, *k*-means clustering
  - Application of SMPC to deep NNs is challenging, due to increased computational costs

# Differential Privacy

*Differential Privacy*

- *Differential privacy* is based on employing obfuscation mechanisms for privacy protection
  - A randomization mechanism $\mathcal{M}(D)$ applies noise ξ to the outputs of a function $f(D)$ to protect the privacy of individual data instances, i.e., $\mathcal{M}(D) = f(D) + \xi$
  - Commonly used randomization mechanisms include Laplacian, Gaussian, and Exponential mechanism
- DP is often implemented in practical applications
- Examples include:
  - 2015: Google, for sharing historical traffic statistics
  - 2016: Apple, for improving its Intelligent Personal Assistant technology
  - 2017: Microsoft, for telemetry in Windows
  - 2020: LinkedIn, for advertiser queries
  - 2020: U.S. Census Bureau, for demographic data

# DP Example

*Differentially Private SGD*

- Consider two databases $D_1$ and $D_2$ that show if a person has diabetes or not
  - The only difference between the two databases is that $D_2$ does not include the last record in $D_1$ (for Bob)
- Let's assume that the databases are publicly available for making queries
  - To protect patient identities, it is not allowed to query the patient names
- However, an adversary can query the sum of the persons with diabetes in the first database (e.g., $f(D_1) = 64$), and the sum in the second database (e.g., $f(D_2) = 63$)
  - Based on the difference $f(D_1) - f(D_2) = 64 - 63 = 1$, the adversary can infer that Bob has diabetes
  - Alternatively, if $f(D_1) = 63$ and $f(D_2) = 63$, the adversary can infer that Bob does not have diabetes

$D_1$ (includes Bob)

| Name | Has Diabetes |
|---|---|
| Don | 1 |
| Monica | 0 |
| … | |
| … | |
| Chris | 1 |
| Bob | 1 |

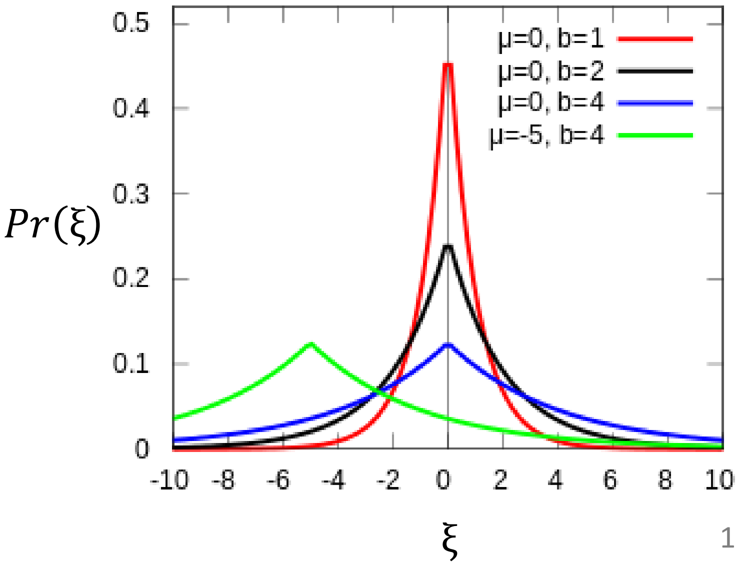
$D_2$ (without Bob)

| Name | Has Diabetes |
|---|---|
| Don | 1 |
| Monica | 0 |
| … | |
| … | |
| Chris | 1 |

# DP Example (cont’d)

*Differentially Private SGD*

- An algorithm that is ***differentially private*** adds noise to the answers for $f(D_1)$ and $f(D_2)$ to make it difficult to infer the information about Bob
  - I.e., a randomization mechanism $\mathcal{M}(D)$ is selected to add noise ξ to the output answers to queries $f(D)$, that is, $\mathcal{M}(D) = f(D) + \xi$
- Additive noise ξ from a Laplacian distribution (shown) is commonly applied
  - E.g., let’s assume a privacy budget $\varepsilon = 0.5$ and let’s sample noise from a Laplacian distribution with $\mu = 0$ and scale $b = 1/\varepsilon = 1/0.5 = 2$
  - 6 random noise samples are: $\xi \in \{-0.13, 2.06, -1.67, -2.49, -0.52,\ 0.37\}$
  - Consider 3 queries by the adversary having the outputs $f(D_1) = 64$ and $f(D_2) = 63$ with added Laplacian noise ξ :
    - ○ $\mathcal{M}(D_1) - \mathcal{M}(D_2) = 63.87 - 65.06 = -1.19$
    - ○ $\mathcal{M}(D_1) - \mathcal{M}(D_2) = 62.33 - 60.51 = 1.82$
    - ○ $\mathcal{M}(D_1) - \mathcal{M}(D_2) = 63.48 - 63.37 = 0.11$
  - Based on the differences between the randomized outputs from the queries for $D_1$ and $D_2$, now it is impossible for the adversary to tell if Bob has diabetes



# DP Mechanism

*Differentially Private SGD*

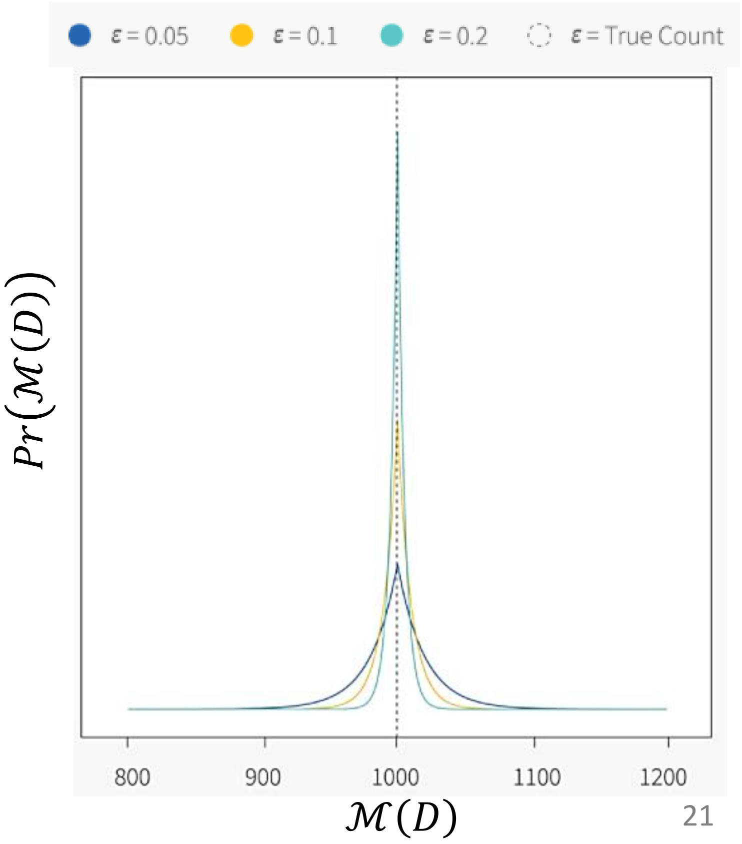
- The important question in DP is: how much noise to add?
  - The amount of noise ξ depends on the data, and it needs to be adjusted
    - E.g., a function $f_1(D)$ that provides the yearly income of people in thousands of dollars would require different level of noise than a function $f_2(D)$ that provides the height in feet
- The ***sensitivity*** of the function $f$ determines how much the output $f(D)$ changes by adding a single data instance
  - Sensitivity is defined as $\Delta f = max\|f(D_1) - f(D_2)\|_1$ for all possible datasets $D_1$ and $D_2$ differing in one data instance, where $\|\cdot\|_1$ denotes $\ell_1$-norm
    - E.g., for the example with medical diabetes records, the sensitivity is $\Delta f = 1$, since the sum of the people with diabetes can change only by 1 when a single input is added
- A Laplacian mechanism that is ε-differentially private adds a Laplacian noise with scale $b = \Delta f / \varepsilon$
- Note that if the privacy budget ε has smaller values, this will result in larger amount of Laplacian noise ξ added to $f(D)$
  - Thus, the noisy outputs $\mathcal{M}(D)$ will reveal less private information about the inputs (i.e., provide better privacy protection), but also the noisy answers to the queries $\mathcal{M}(D)$ will be less accurate

# DP with Laplacian Randomization

- The figure shows the probability distributions of the outputs $\mathcal{M}(D)$ for three different levels of Laplacian noise with $\varepsilon \in \{0.05, 0.1, 0.2\}$
  - The true output value is $f(D) = 1{,}000$
  - Larger values of $\varepsilon$ have distributions that are tighter around the true value of $f(D) = 1{,}000$ in the figure, and hence are more accurate, but leak more privacy
- A mechanism $\mathcal{M}(D)$ is $\varepsilon$-differentially private if for all databases $D_1$ and $D_2$ that differ by at most one instance, and for any subset of outputs $S$:

$$Pr(\mathcal{M}(D_1) \in S) \leq e^{\varepsilon}\, Pr(\mathcal{M}(D_2) \in S)$$

  - In other words, $\varepsilon$-differential privacy ensures that the probabilities of any two outputs $\mathcal{M}(D_1)$ and $\mathcal{M}(D_2)$ differ by at most $e^{\varepsilon}$
  - E.g., for $\varepsilon = 0.05$, $Pr\big(\mathcal{M}(D_1)\big)/Pr\big(\mathcal{M}(D_2)\big)$ is at most $e^{0.05} = 1.05$
  - Smaller $\varepsilon$ ensures more similar outputs $\mathcal{M}(D_1)$ and $\mathcal{M}(D_2)$, and provides higher levels of privacy



# DP with Gaussian Randomization

*Differentially Private SGD*

- There are other DP mechanisms besides the Laplacian mechanism, that are more suitable for some applications
- The ***Gaussian mechanism*** adds Gaussian noise instead of Laplacian noise, and the level of noise is based on the $\ell_2$-norm sensitivity, instead of $\ell_1$-norm
- A Gaussian mechanism is $(\varepsilon, \delta)$-differentially private if for all databases $D_1$ and $D_2$ that differ by at most one instance, and for any subset of outputs $S$:

$$Pr(\mathcal{M}(D_1) \in S) \leq e^{\varepsilon}\, Pr(\mathcal{M}(D_2) \in S) + \delta$$

- The $(\varepsilon, \delta)$-differential privacy that is provided by the Gaussian mechanism introduces the probability parameter $\delta$
  - Informally, $(\varepsilon, \delta)$-differential privacy is guaranteed with probability $1 - \delta$
  - E.g., for $\delta = 0.05$, the method is $\varepsilon$-differentially private with 95% probability
- The Gaussian mechanism is therefore weaker than the Laplacian mechanism, since it allows scenarios when the privacy cannot be guaranteed

# DP in Machine Learning

*Differentially Private SGD*

- Training ML models can be considered an extension of the previous example on querying databases
  - I.e., ML models use data to learn a function, which is afterward used for prediction
- The datasets for training ML models often contain sensitive information (e.g., medical records, personal information), so it is important to provide privacy guarantees
  - On the other hand, we know that ML models can memorize the training data, which can be exploited by adversaries to recover information about the data from a trained model
- The challenge is: how to extract enough information from data to train accurate ML models without revealing the data

# DP in Machine Learning

*Differential Privacy*

- In ML, DP is achieved by adding noise to:
  - *Model parameters*
    - Several works applied DP to conventional ML methods
    - Differentially private SGD (Abadi, 2016) clips and adds noise to the gradients of deep NNs during training
      - This reduces the memorization of individual input instances by the model
    - The approaches that apply obfuscation to the model parameters via DP are also referred to as differentially private ML
  - *Model outputs*
    - PATE (Private Aggregation of Teacher Ensembles) approach (Papernot, 2018) employs an ensemble of models trained on disjoint subsets of the training data, called teacher models
    - Noise is added to the outputs of the teacher models, and the aggregated outputs are used to train another model, called student model
  - *Training data*
    - Obfuscation of training data in ML has been also investigated in several works

# DP in Machine Learning

- DP is typically applied in a ***centralized learning setting***, where the data and model are at the same location
  - In this scenario, all data is gathered in one central location for model training
  - E.g., MLaaS typically requires that the users upload their data to a cloud-based server for training a model
- Recently, DP has also been applied in a ***distributed learning setting***, where the data are kept at separate locations from the model
  - DP-FedAvg (McMahan, 2018) is applied to federated learning
  - It introduced the Federated Averaging algorithm to limits the contributions by the data from individual users to the learning model

# Distributed Learning

*Distributed Learning*

- *Distributed learning* allows multiple parties to train a global model without releasing their private data
- Some form of aggregation is applied to the local updates of the model parameters by the users in distributed learning to create a global model
  - E.g., averaging is one common form of aggregation
- Federated learning is the most popular distributed learning scheme

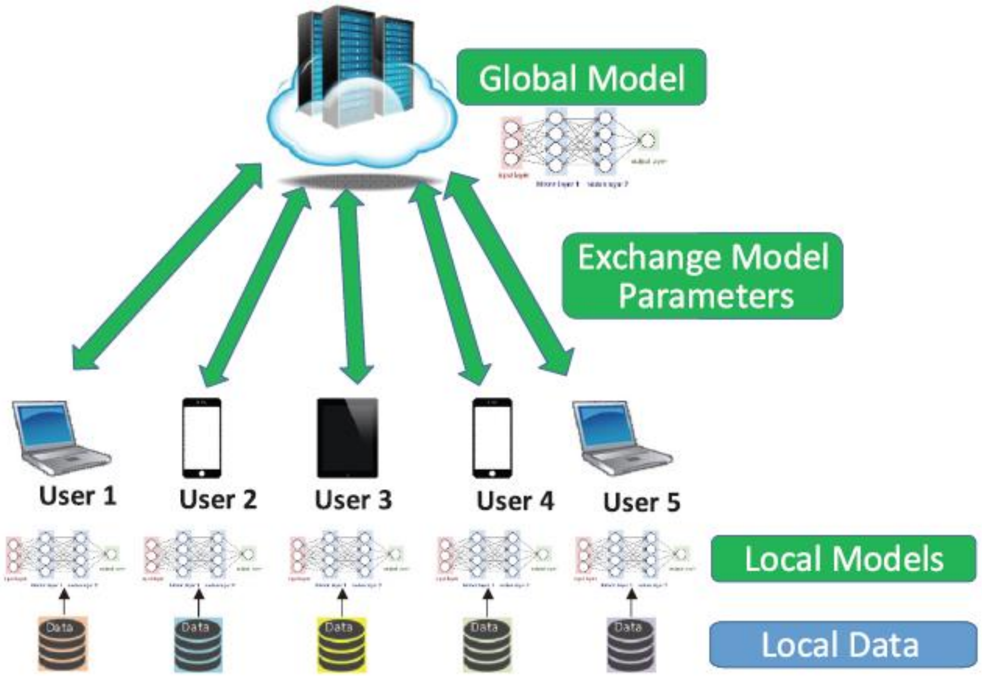


Figure form: Liu et al. (2020) When Machine Learning Meets Privacy: A Survey and Outlook

# Distributed Learning

*Distributed Learning*

- *Federated learning* or *collaborative learning* – learn one global model using data stored at multiple locations (e.g., remote devices)
  - The data are processed locally, and used to update the model
    - The data do not leave the remote devices, remains private
  - The central server aggregates the updates and creates the global model
- *Decentralized Peer-to-Peer (P2P) learning* – the remote devices communicate and exchange the updates directly, without a central server
  - Removes the need to send updates to a potentially untrusted central server
- *Split learning* – each remote device is used to train several layers of the global model, and send the outputs to a central server
  - The remote devices can train the initial layers of a DNN, and the central server can train the final layers
    - The gradient is back-propagated from the central server to each user to sequentially complete the back-propagation through all layers of the model
  - The devices send the intermediate layers outputs, rather than model parameters
  - Split learning is more common for IoT devices with limited computational resources

# ML-Specific Techniques

- In the lecture on privacy attacks in ML, we mentioned that overfitting is one of the reasons for information leakage
- ***Regularization techniques*** in ML can therefore be used to reduce overfitting, as well as a defense strategy
  - Different regularization techniques in NNs include:
    - Explicit regularization: dropout, early stopping, weight decay
    - Implicit regularization: batch normalization
- Other ML-specific techniques include:
  - Dimensionality reduction – removing inputs with features that occur rarely in the training set
  - Weight-normalization – rescaling the weights of the model during training
  - Selective gradient sharing – in federated learning, the users share a fraction of the gradient at each update

# References

1. Liu et al. (2020) When Machine Learning Meets Privacy: A Survey and Outlook (link)
2. Rigaki and Carcia (2021) A Survey of Privacy Attacks in Machine Learning (link)
3. Cristofaro (2020) An Overview of Privacy in Machine Learning (link)
4. Borealis AI Tutorial #13: Differential Privacy II: Machine Learning and Data Generation (link)
5. Davide Testuggine and Ilya Mironov blog: Differential Privacy Series Part 1- DP-SGD Algorithm Explained (link)
6. Joseph P. Near and Chiké Abuah, Programming Differential Privacy – Chapter III: Differential Privacy (link)

University of Idaho
College of Engineering

# DIFFERENTIAL PRIVACY

JASON STARACE

# DIFFERENTIAL PRIVACY

## TOPICS

- What is Differential Privacy
  - History of
  - In English
  - Formally
- Fundamentals
- Laplace Randomization
- Approximate Differential Privacy
- Privacy Loss as a random variable
- Gaussian/Binomial noise
- A real use case
  - Deep Learning with Differential Privacy

# WHAT IS DIFFERENTIAL PRIVACY

## HISTORY

- The definition of differential privacy came from a long line of work applying algorithmic ideas to the study of privacy [1], culminating with the work of:

Cynthia Dwork

Source: Harvard Radcliffe Institute

Frank McSherry

Source: github.com/frankmcsherry

Kobbi Nissim

Source: CRCS

Adam Smith

Source: Boston University

With their 2006 paper titled “Calibrating Noise to Sensitivity in Private Data Analysis”.  The paper has since been revised and updated over the years with responses and additional information

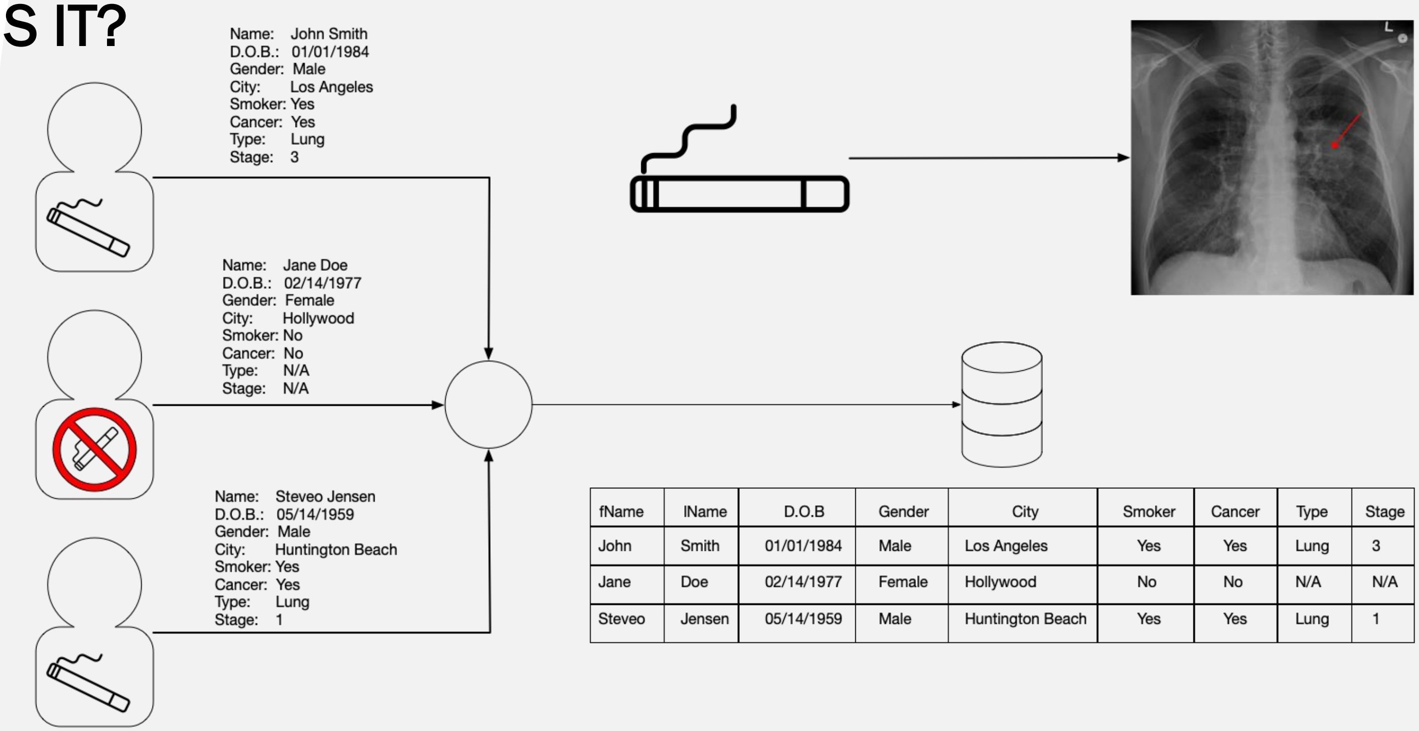
# WHAT IS DIFFERENTIAL PRIVACY

## SO WHAT IS IT?

- English language definition:
  - “The outcome of any analysis is essentially equally likely, independent of whether any individual joins, or refrains from joining, the dataset” [4]

# WHAT IS DIFFERENTIAL PRIVACY

## SO WHAT IS IT?



Name: John Smith
D.O.B.: 01/01/1984
Gender: Male
City: Los Angeles
Smoker: Yes
Cancer: Yes
Type: Lung
Stage: 3

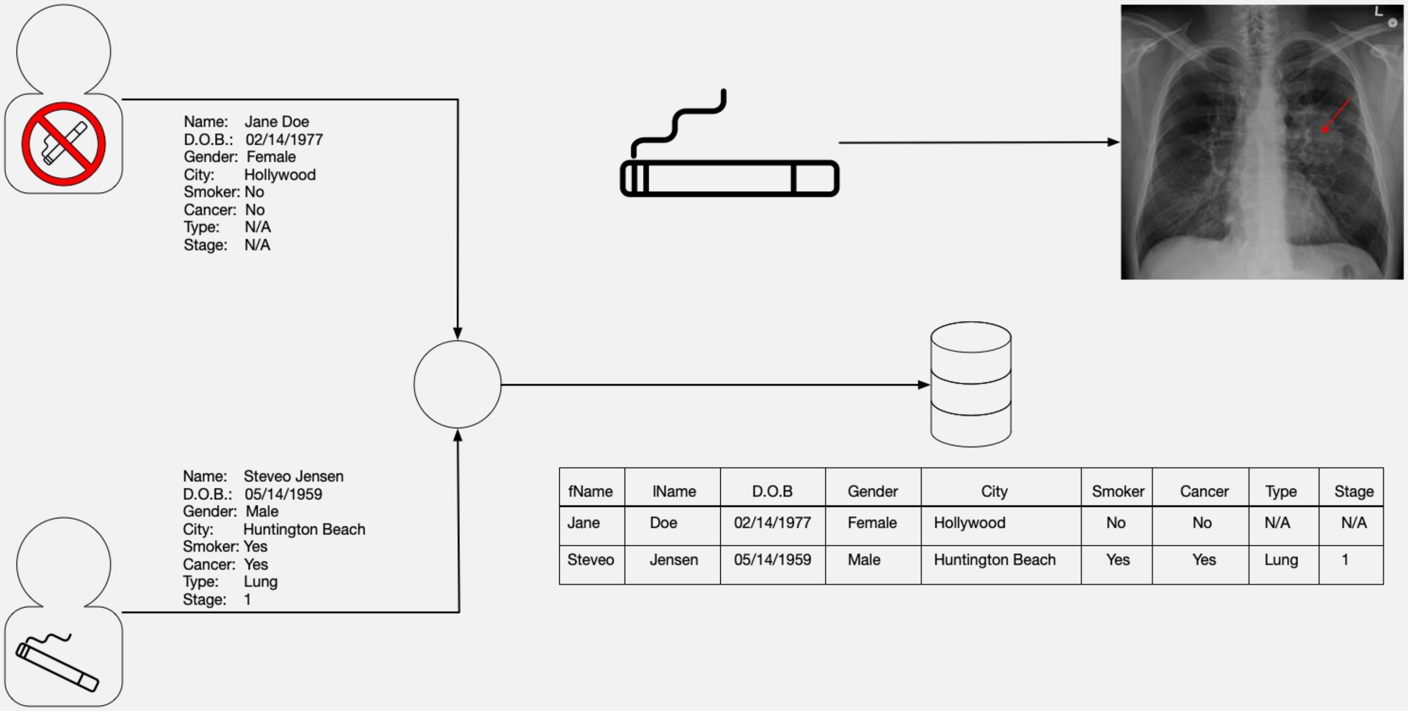
Name: Jane Doe
D.O.B.: 02/14/1977
Gender: Female
City: Hollywood
Smoker: No
Cancer: No
Type: N/A
Stage: N/A

Name: Steveo Jensen
D.O.B.: 05/14/1959
Gender: Male
City: Huntington Beach
Smoker: Yes
Cancer: Yes
Type: Lung
Stage: 1

| fName | lName | D.O.B | Gender | City | Smoker | Cancer | Type | Stage |
|---|---|---|---|---|---|---|---|---|
| John | Smith | 01/01/1984 | Male | Los Angeles | Yes | Yes | Lung | 3 |
| Jane | Doe | 02/14/1977 | Female | Hollywood | No | No | N/A | N/A |
| Steveo | Jensen | 05/14/1959 | Male | Huntington Beach | Yes | Yes | Lung | 1 |

# WHAT IS DIFFERENTIAL PRIVACY

## SO WHAT IS IT?



Name: Jane Doe
D.O.B.: 02/14/1977
Gender: Female
City: Hollywood
Smoker: No
Cancer: No
Type: N/A
Stage: N/A

Name: Steveo Jensen
D.O.B.: 05/14/1959
Gender: Male
City: Huntington Beach
Smoker: Yes
Cancer: Yes
Type: Lung
Stage: 1

| fName | lName | D.O.B | Gender | City | Smoker | Cancer | Type | Stage |
|---|---|---|---|---|---|---|---|---|
| Jane | Doe | 02/14/1977 | Female | Hollywood | No | No | N/A | N/A |
| Steveo | Jensen | 05/14/1959 | Male | Huntington Beach | Yes | Yes | Lung | 1 |

# WHAT IS DIFFERENTIAL PRIVACY

## KEY TERMS [5]

$x, x', y, y'$ - (Adjacent) Datasets that are nearly identical. The first dataset is slightly larger by 1 record.
$\epsilon$ - Privacy loss - Some value that is $\geq 0$
$\mathcal{M}$ - Algorithm (mechanism) operating on the dataset

# WHAT IS DIFFERENTIAL PRIVACY

## FORMAL DEFINITION [4][5]

$\mathcal{M}$ gives $\epsilon$-differential privacy if for all pairs of data sets $x, y$ differing in the data of one person, and all events $\mathcal{S}$

$$Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} \mathrm{Pr}[\mathcal{M}(y) \in \mathcal{S}]$$

Randomness is introduced by $\mathcal{M}$

# FUNDAMENTALS

## PROPERTIES OF ALL DIFFERENTIALLY PRIVATE ALGORITHMS

[Dwork et al.]

- **Immunity to Auxiliary Information.** Differential privacy makes no reference to an input distribution.
- **Postprocessing.** Anything derived from the output of a differentially private algorithm is itself differentially private and therefore suffers no additional privacy loss.
- **Composition.** Differentially private algorithms *compose,* in the sense that when several differentially private algorithms are run “independently”, then the joint output of all the algorithms is still differentially private.
- **Group Privacy.** Differential privacy with respect to changes of an individual’s data implies differential privacy with respect to changes in the data of small sets of individuals
  - if $\epsilon$ for an individual, then $k\epsilon$ for groups of size $k$

# RANDOMIZATION METHODS

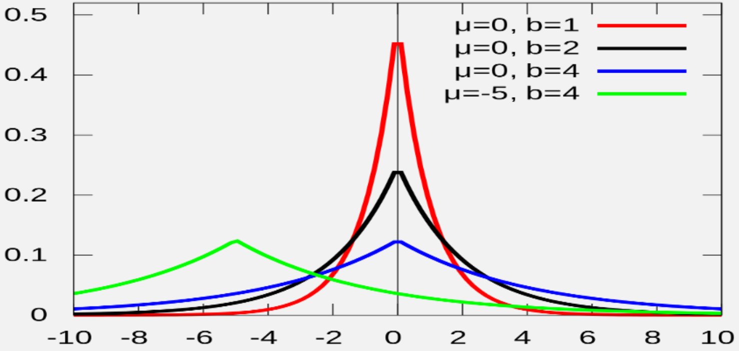
## LAPLACE MECHANISM

[Dwork et. Al][5]

$$\Delta_1 = \max_{adj\ x,y} |f(x) - f(y)|$$

$$M(x) = f(x) + Y, where\ Y \sim Lap(\Delta_1/\varepsilon)$$

Laplace Probability Distribution Function



Source: Wikipedia

# APPROXIMATE DIFFERENTIAL PRIVACY

[5]

$\mathcal{M}$ gives $(\epsilon, \delta)$-differential privacy if for all pairs of data sets $x, y$ differing in the data of one person, and all events $\mathcal{S}$

$$Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

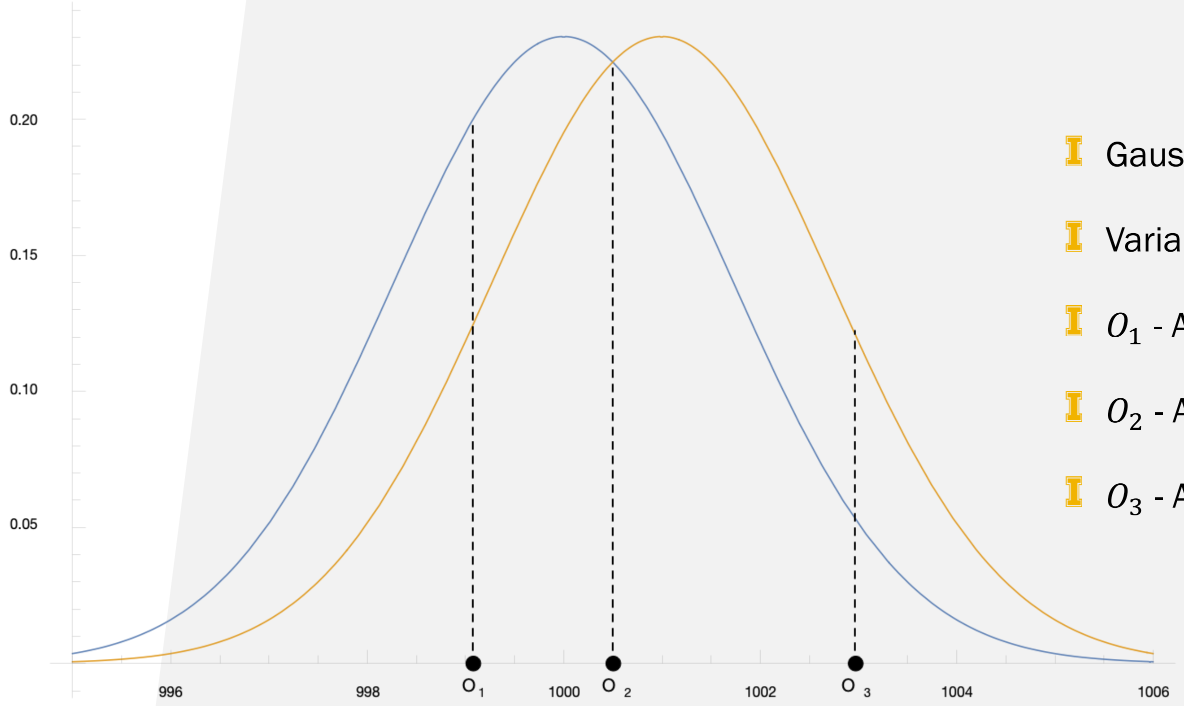
Allows for use of Gaussian/Binomial noise

| Reason | A/D | Details | Impact |
|---|---|---|---|
| Depends on L2 Sensitivity | Advantage | $\left\|\|f(x) - f(y)\right\|\|_2$ | Improve by a factor of $\approx \sqrt{k}$ |
| Advanced Composition | Advantage | $privacy\ loss \leq \sqrt{k \log\left(\frac{1}{\delta}\right)}\,\epsilon$ with a probability $\geq 1 - \delta$ | Privacy loss as mentioned rather than $k\epsilon$ |
| Concentrated DP (CDP) & others | Advantage | | Privacy loss rv is subgaussian |

# PRIVACY LOSS RANDOM VARIABLE

[5]&[6]

$$privacyLoss\,(c) = \ln\left[\frac{Pr[\mathcal{M}(x) = c]}{Pr[\mathcal{M}(y) = c]}\right]$$



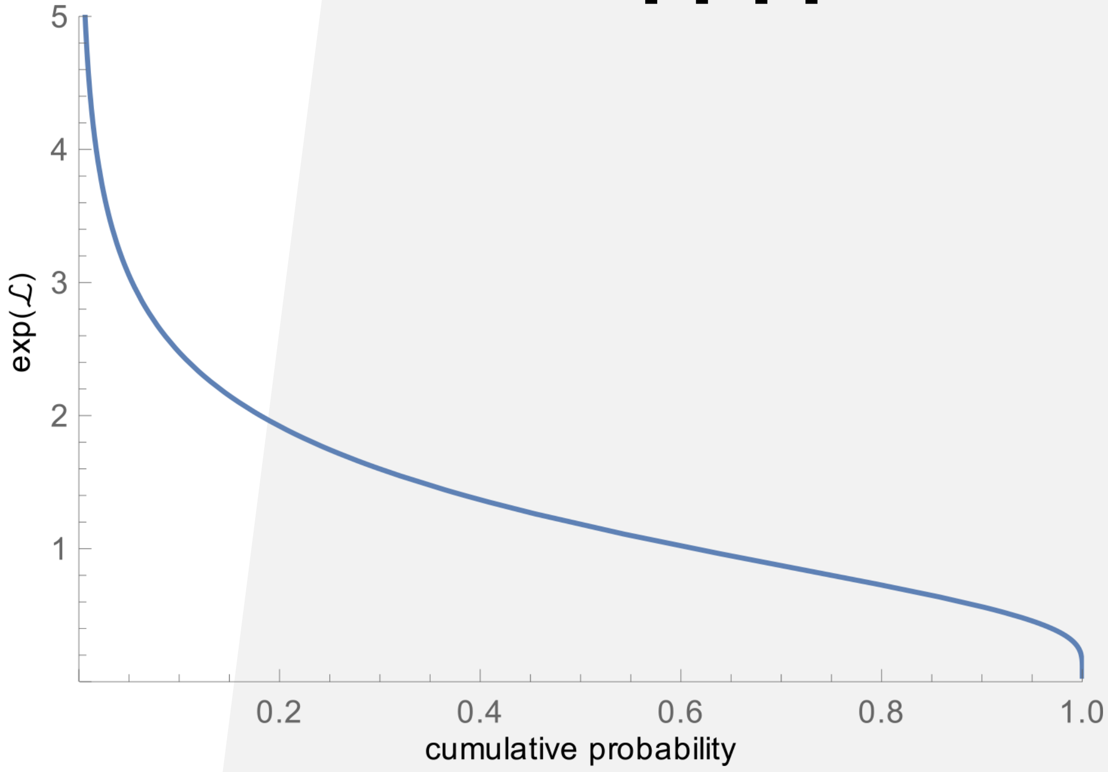
- Gaussian noise
- Variance $\sigma^2 = 2$
- $O_1$ - Attacker suspects real is $D_1$
- $O_2$ - Attacker is confused
- $O_3$ - Attacker is tricked thinking $D_2$ is the real database

# PRIVACY LOSS RANDOM VARIABLE

## A FORMAL DEFINITION [5]&[6]

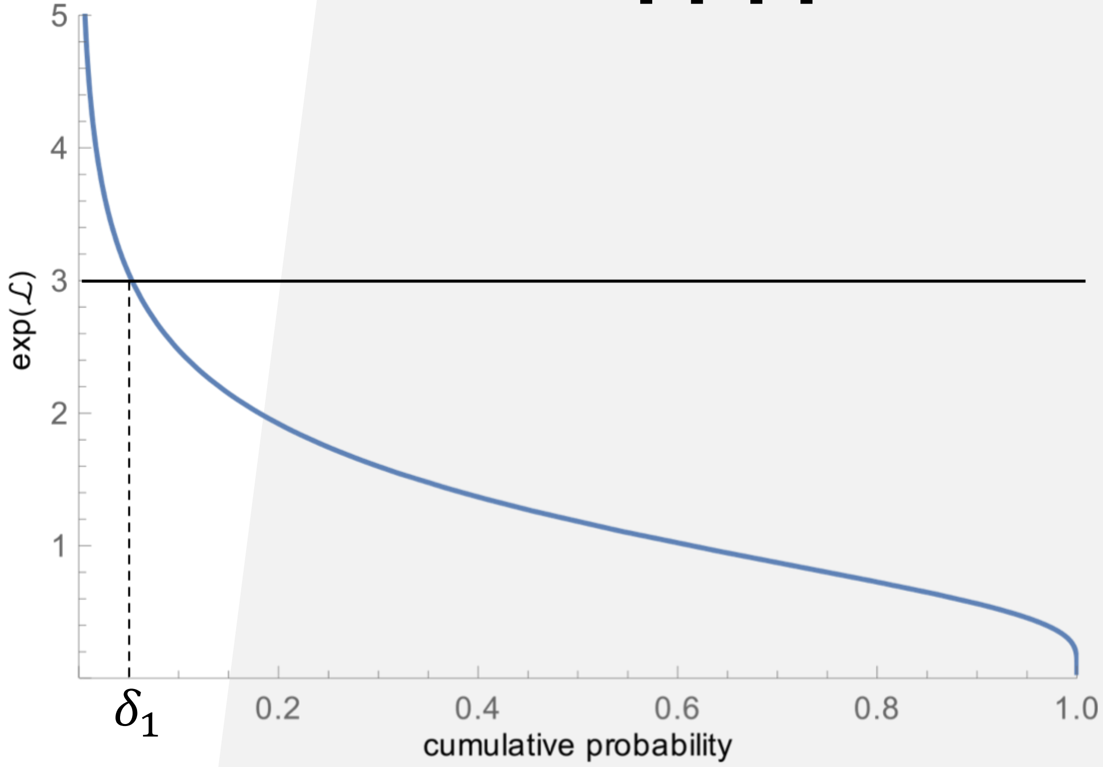
$$privacyLoss\ (c) = \ln\left[\frac{Pr[\mathcal{M}(x) = c]}{Pr[\mathcal{M}(y) = c]}\right]$$

- All possible events $\mathcal{O} = \mathcal{M}(D_1)$ in order of what benefits the attacker first
- $\mathcal{L} \rightarrow \mathcal{L}_{D_1,D_2}(\mathcal{O})$
- Since we're more interested in $e^{\epsilon}$ we will graph against $\exp(\mathcal{L})$



# PRIVACY LOSS RANDOM VARIABLE

## A FORMAL DEFINITION [5]&[6]



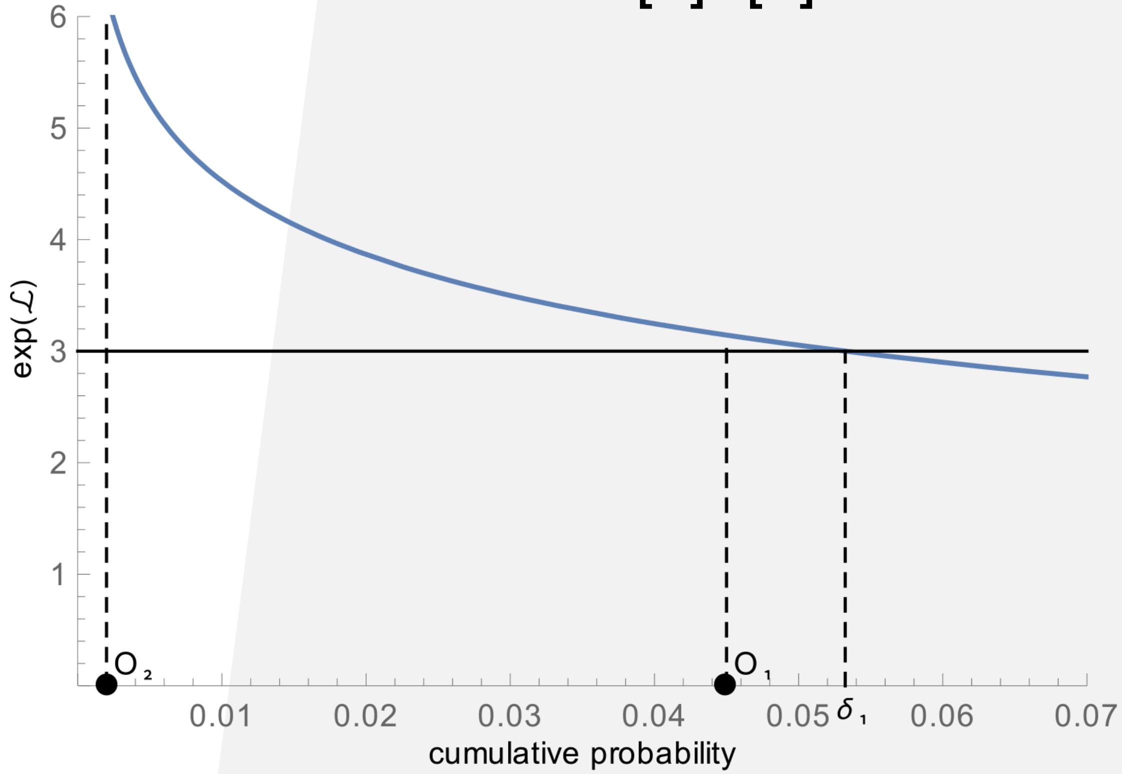
$$privacyLoss\ (c) = \ln\left[\frac{Pr[\mathcal{M}(x) = c]}{Pr[\mathcal{M}(y) = c]}\right]$$

- All possible events $\mathcal{O} = \mathcal{M}(D_1)$ in order of what benefits the attacker first
- $\mathcal{L} \rightarrow \mathcal{L}_{D_1,D_2}(\mathcal{O})$
- Since we're more interested in $e^{\epsilon}$ we will graph against $\exp(\mathcal{L})$
- Pick an arbitrary $\epsilon = \ln(3)$
- $\delta$ Represents something terrible happening
- How likely is it for $e^{\mathcal{L}}$ to be above $e^{\epsilon} = 3$
    - Measure the width -> $(\ln(3), \delta_1) - DP, \delta_1 \approx .054$

# PRIVACY LOSS RANDOM VARIABLE

## A FORMAL DEFINITION [5]&[6]

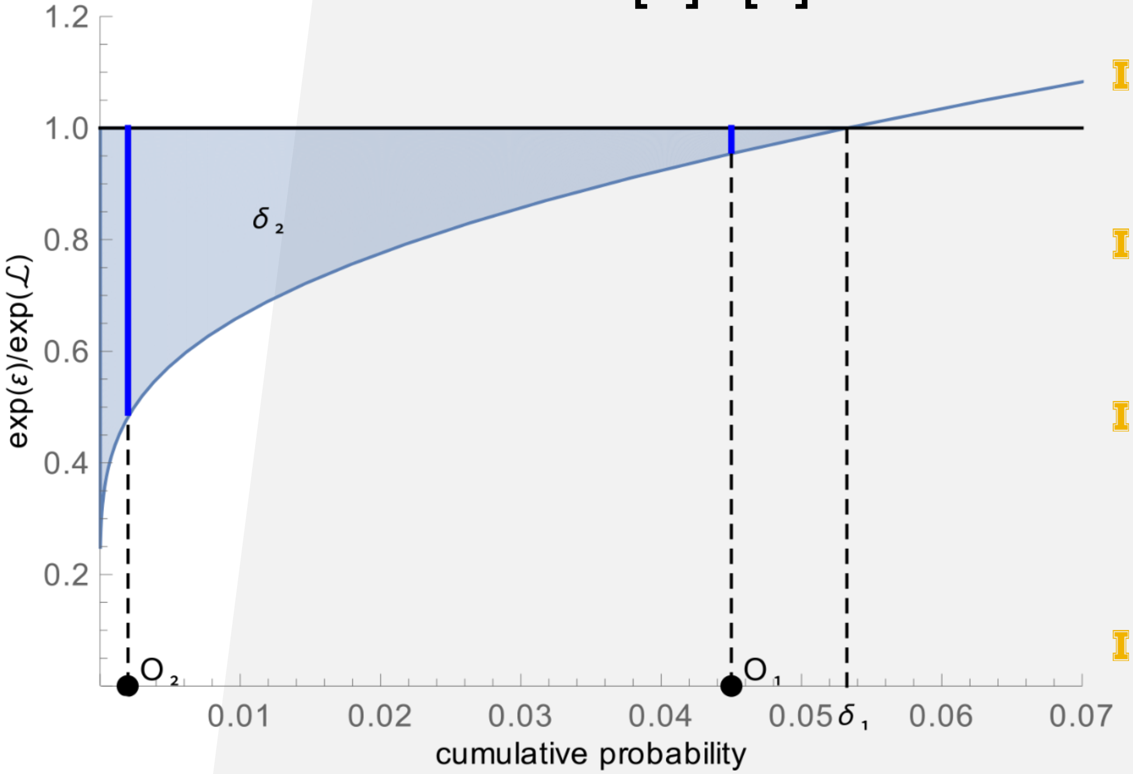
$$privacyLoss\,(c) = \ln\left[\frac{Pr[\mathcal{M}(x) = c]}{Pr[\mathcal{M}(y) = c]}\right]$$



- Returning $O_1$ is not great but not terrible $e^{\mathcal{L}} > e^{\epsilon}$
- Returning $O_2$ is a lot worse, it's obvious it leaks more
- $\delta$ doesn't account for this leakage
- Resolve this by adding weights
  - $\approx 1$ to the very bad events
  - $\approx 0$ to the 'not so' bad events
- But how?

# PRIVACY LOSS RANDOM VARIABLE

## A FORMAL DEFINITION [5]&[6]



$$privacyLoss\ (c) = \ln\left[\frac{Pr[\mathcal{M}(x) = c]}{Pr[\mathcal{M}(y) = c]}\right]$$

- Take the inverse of the curve
  - Bad events approach 0
- Normalize the curve using ratio $\frac{e^{\varepsilon}}{e^{\mathcal{L}}}$
  - Events not too bad close 1
- This is the mass of all possible bad events, weighted by how likely they are and how bad they are.
  - $(\ln(3), \delta_2) - DP\ with\ \delta_2 \approx 0.011$
- This is a tighter characterization of $\delta$ and not used in the typical definition of $(\epsilon, \delta)$

# GAUSSIAN MECHANISM

## A FORMAL DEFINITION [5]

$$|Privacy\ Loss| \leq \epsilon \implies \left| \ln \frac{e^{-z^2/2\sigma^2}}{e^{-(z+\Delta f)^2/2\sigma^2}} \right| \leq \epsilon$$

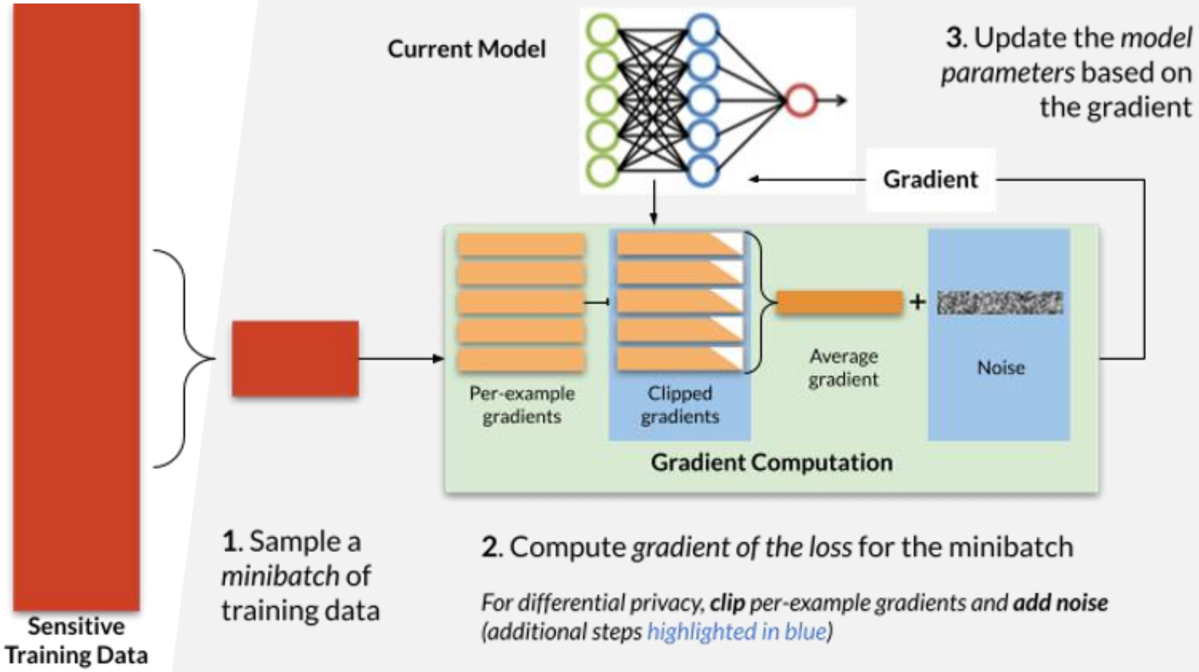
$$To\ achieve\ (\epsilon, \delta) - DP, set\ \sigma = \Delta f \sqrt{2\ \ln 1/\delta/\epsilon}$$

When outside range -> $|z| < \sigma\left(1 - \frac{\epsilon}{2}\right) \longrightarrow |Privacy\ Loss|\ at\ most\ 2\epsilon$

$$PR[|Privacy\ Loss| \geq t\epsilon] \leq PR\left[|z| > \sigma\left(t - \frac{\epsilon}{2}\right)\right] \approx \frac{1}{t} e^{-t^2/2}$$

# DP AND DEEP LEARNING

## DP-SGD [7]

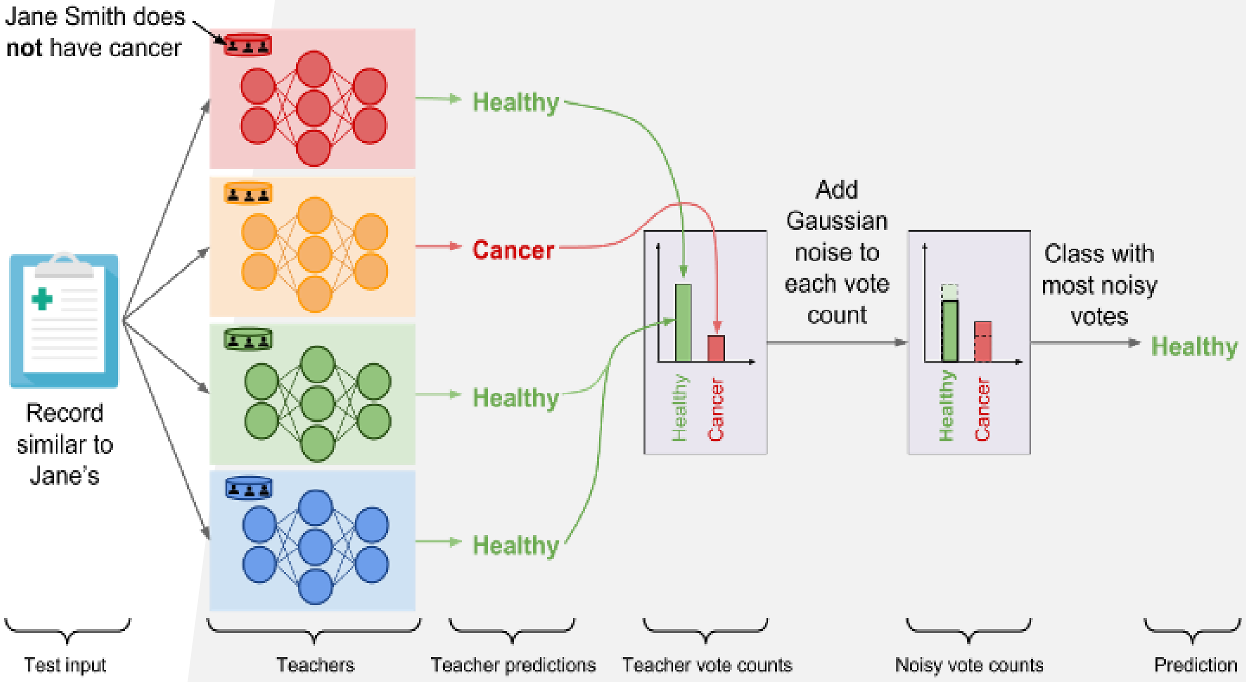


Current Model
3. Update the *model parameters* based on the gradient
Gradient
Per-example gradients
Clipped gradients
Average gradient
+
Noise
Gradient Computation
Sensitive Training Data
1. Sample a *minibatch* of training data
2. Compute *gradient of the loss* for the minibatch
For differential privacy, **clip** per-example gradients and **add noise** (additional steps highlighted in blue)

- Gradients computed per-example as opposed to computed on the average
- Gradients are clipped to control sensitivity
- Spherical Gaussian noise $b_t$ is added to their sum
- Update step can be rewritten as:
  - $\theta_{t+1} \leftarrow \theta_t - \eta \cdot (\nabla_t + b_t)$
- Adding noise during training can hurt accuracy

# DP AND DEEP LEARNING

## MODEL AGNOSTIC PRIVATE LEARNING [7]



- Consider a multi-class classification problem.
- Split training data into k disjoint subsets of equal size
- Train independent models $(\theta_1(x) \ldots \theta_k(x))$
- Compute a private histogram over the set of k predictions.
- Add noise
- Select histogram with highest count

University of Idaho

College of Engineering

# Q & A

# REFERENCES

[1] Differential privacy. [Online]. Available: https://privacytools.seas.harvard.edu/differential-privacy (visited on 04/12/2023)

[2] Dwork, C., McSherry, F., Nissim, K., Smith, A. (2006). Calibrating Noise to Sensitivity in Private Data Analysis. In: Halevi, S., Rabin, T. (eds) Theory of Cryptography. TCC 2006. Lecture Notes in Computer Science, vol 3876. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11681878_14

[3] [Dwork et al.] Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2017). Calibrating Noise to Sensitivity in Private Data Analysis. Journal of Privacy and Confidentiality, 7(3), 17–51. https://doi.org/10.29012/jpc.v7i3.405

[4] Institute for Advanced Study. (2016, November 14). *The Definition of Differential Privacy - Cynthia Dwork* [Video]. YouTube. https://www.youtube.com/watch?v=lg-VhHlztqo

[5] Mathematical Picture Language. (2021, October 5). *Cynthia Dwork | Oct 5,2021 | Differential Privacy:The Mathematical Bulwark* [Video]. YouTube. https://www.youtube.com/watch?v=W--U-O3h_is

[6] Desfontaines, D. (2020, March 6). *The privacy loss random variable - Ted is writing things*. https://desfontain.es/privacy/privacy-loss-random-variable.html

[7] Papernot, N. (2022). How to deploy machine learning with differential privacy | NIST. *NIST*. https://www.nist.gov/blogs/cybersecurity-insights/how-deploy-machine-learning-differential-privacy

# DEEP LEARNING WITH DIFFERENTIAL PRIVACY

LU CAI

# OUTLINE

- Background
- Approach
- Results – MNIST
- Conclusion

# BACKGROUND

## DIFFERENTIAL PRIVACY AND DEEP LEARNING

- Differential privacy is a privacy framework that aims to protect sensitive information while still allowing for useful data analysis.
- In practice, differential privacy involves adding noise to data before it is analyzed, so that the true values of individual data points are obscured. One of the key challenges in implementing differential privacy is balancing privacy with the usefulness of the data.
- Deep neural networks are highly expressive models that can potentially memorize individual training examples. Deep learning with differential privacy is an emerging field that combines the power of deep neural networks with the privacy guarantees of differential privacy. The goal is to develop machine learning algorithms that can analyze sensitive data while preserving the privacy of individuals in the data.

# BACKGROUND

## DIFFERENTIAL PRIVACY

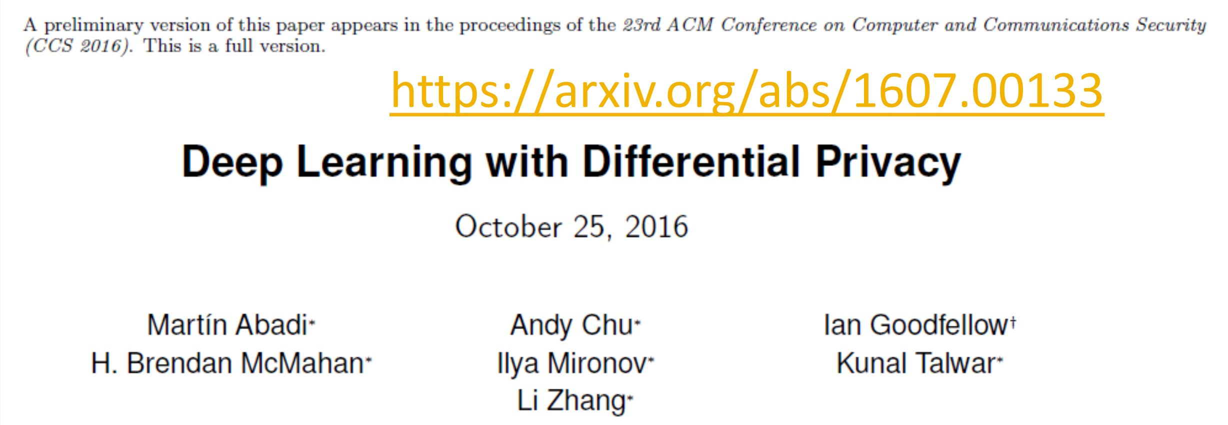
- The formal definition of (ε, δ)-differential privacy:
  - A randomized mechanism M: D → R satisfies ε-differential privacy if for any two adjacent inputs d, d' ∈ D and for any subset of outputs S ⊆ R,  it holds that:

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta.$$

  - ε is a parameter that determines the strength of the privacy guarantee provided by a differentially private mechanism. A smaller value of ε corresponds to a stronger privacy guarantee.
  - δ is the probability that allows for plain ε-differential privacy broken.

# APPROACH

## DIFFERENTIAL PRIVATE TRAINING OF NEUTRAL NETWORK



A preliminary version of this paper appears in the proceedings of the *23rd ACM Conference on Computer and Communications Security (CCS 2016)*. This is a full version.

https://arxiv.org/abs/1607.00133

### Deep Learning with Differential Privacy

October 25, 2016

Martín Abadi[*] Andy Chu[*] Ian Goodfellow[†]
H. Brendan McMahan[*] Ilya Mironov[*] Kunal Talwar[*]
Li Zhang[*]

- A differentially private stochastic gradient descent (SGD) algorithm
- The moments accountant
- Hyperparameter tuning

# APPROACH

## A DIFFERENTIALLY PRIVATE STOCHASTIC GRADIENT DESCENT (SGD) ALGORITHM

- Compute the gradient for a random subset of examples (batch)
- Clips the $l_2$ norm
- Compute the average
- Add noise
- Take a step in the opposite direction of this average noisy gradient

Note: The authors perform the computation in batches, then group several batches into a lot for adding noise

# APPROACH

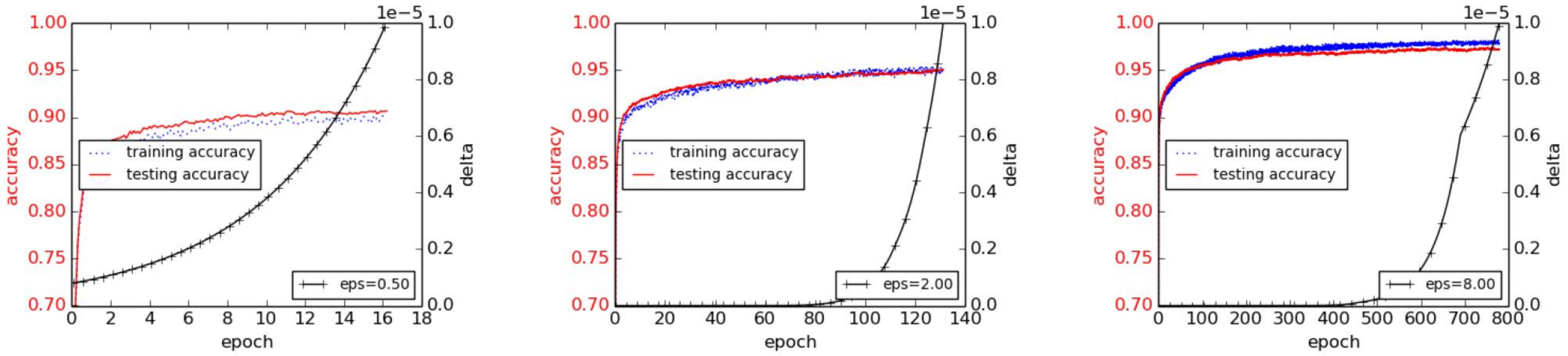
## PRIVACY ACCOUNTING AND HYPERPARAMETER TUNING

- For differentially private SGD, to compute the overall privacy cost of the training, an “accountant" procedure is implemented,  which computes the privacy cost at each access to the training data, and accumulates this cost as the training progresses.

- Hyperparameters can be tuned to balance privacy, accuracy, and performance.  The model accuracy is more sensitive to training parameters such as batch size and noise level than to the structure of a neural network.

# RESULTS - MNIST

- Experiments on standard MNIST dataset for handwritten digit recognition consisting of 60,000 training examples and 10,000 testing examples
- Baseline Model accuracy: 98.3%

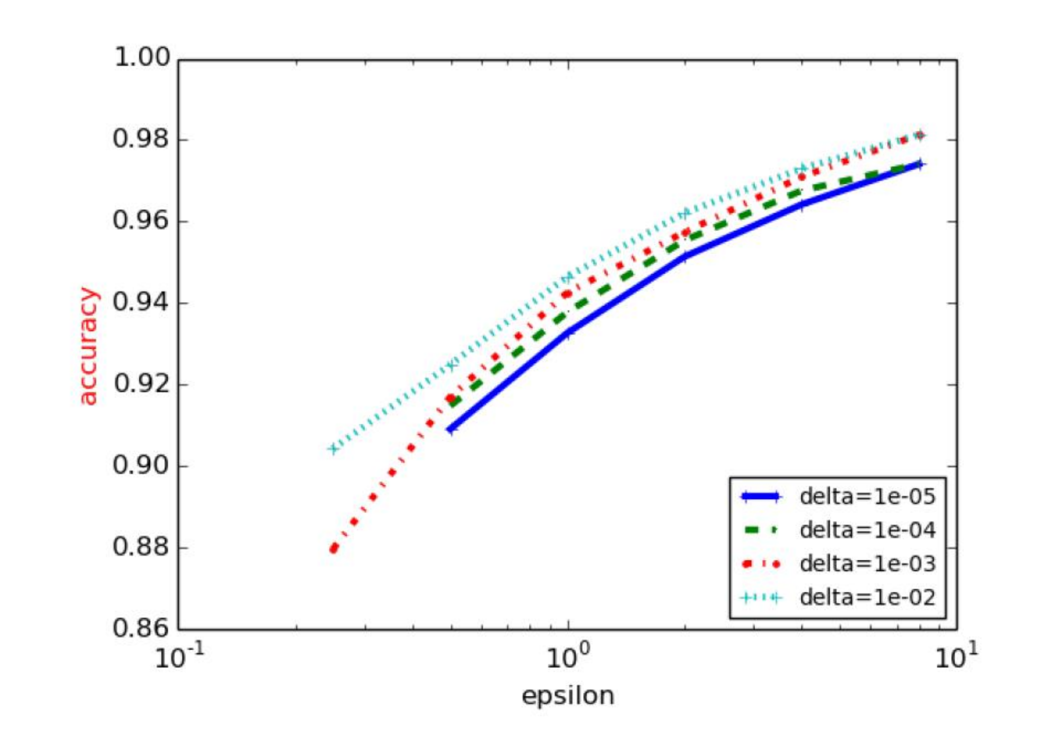


(1) Large noise

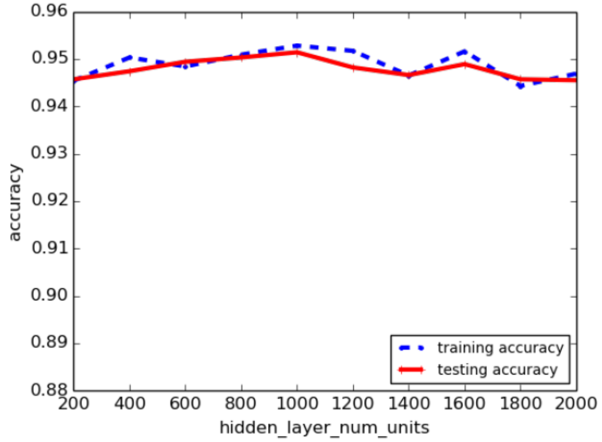
(2) Medium noise

(3) Small noise

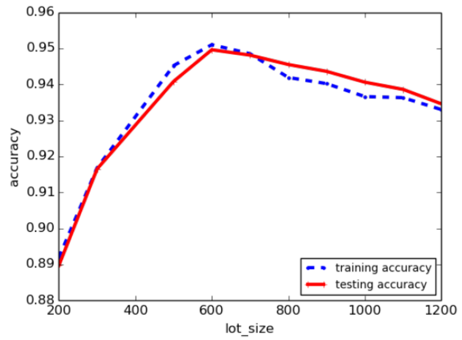
# RESULTS - MNIST



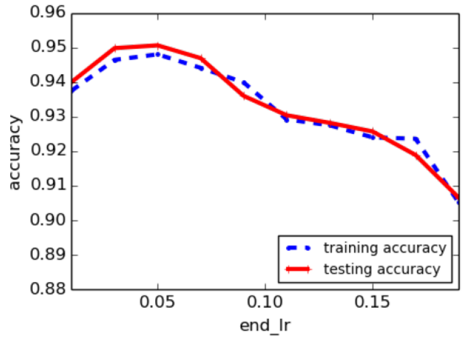
# RESULTS - MNIST



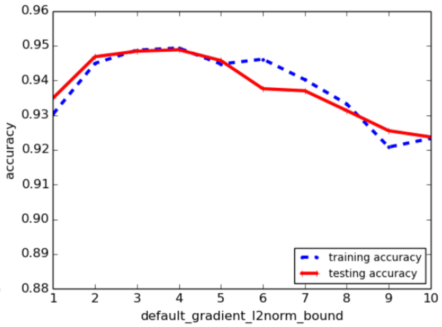
variable hidden units



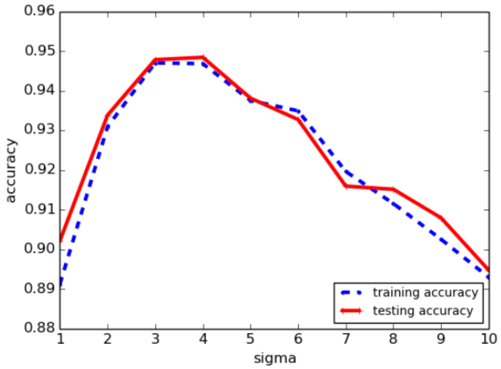
variable lot size



variable learning rate



variable gradient clipping norm



variable noise level

The model accuracy is more sensitive to training parameters such as learning rate and noise level than to the structure of a neural network.

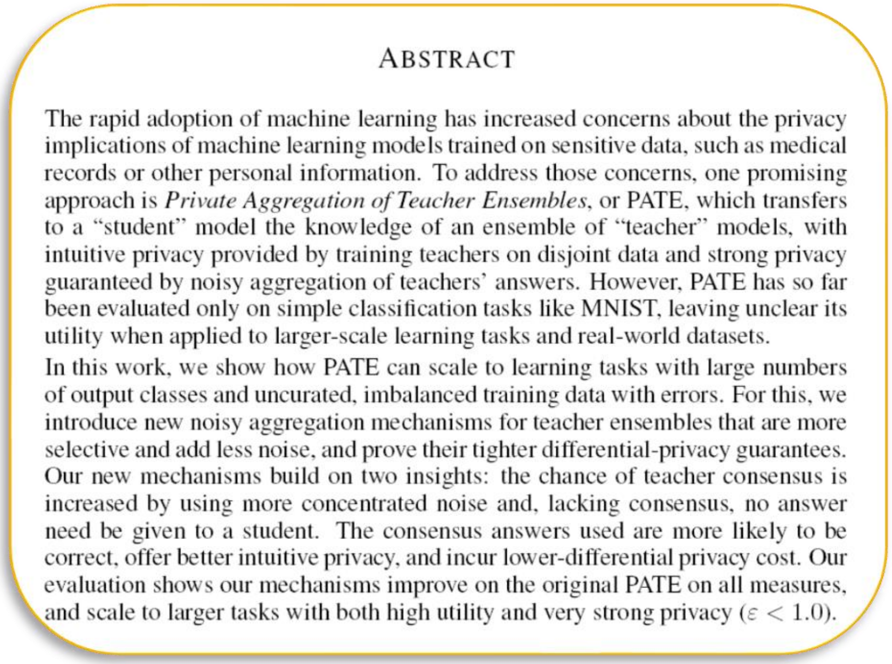
# CONCLUSION

- Deep learning with differential privacy is an approach to machine learning that aims to protect the privacy of sensitive data while still allowing for effective learning.
- The algorithms of the paper are based on a differentially private version of stochastic gradient descent (SGD) and reach 97% training accuracy with (8; $10^{-5}$)-differential privacy.

University of Idaho
College of Engineering

# Scalable Private Learning with PATE

## Private Aggregation of Teacher Ensembles



ABSTRACT

The rapid adoption of machine learning has increased concerns about the privacy implications of machine learning models trained on sensitive data, such as medical records or other personal information. To address those concerns, one promising approach is *Private Aggregation of Teacher Ensembles*, or PATE, which transfers to a "student" model the knowledge of an ensemble of "teacher" models, with intuitive privacy provided by training teachers on disjoint data and strong privacy guaranteed by noisy aggregation of teachers' answers. However, PATE has so far been evaluated only on simple classification tasks like MNIST, leaving unclear its utility when applied to larger-scale learning tasks and real-world datasets.

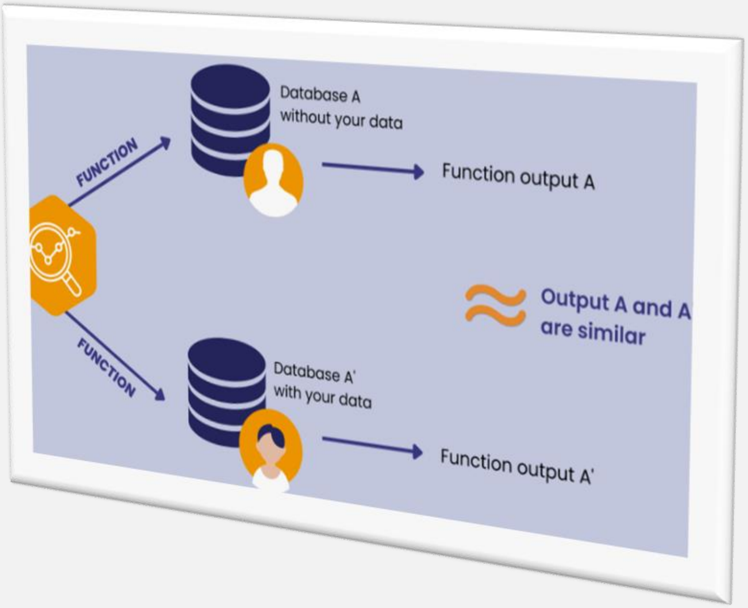
In this work, we show how PATE can scale to learning tasks with large numbers of output classes and uncurated, imbalanced training data with errors. For this, we introduce new noisy aggregation mechanisms for teacher ensembles that are more selective and add less noise, and prove their tighter differential-privacy guarantees. Our new mechanisms build on two insights: the chance of teacher consensus is increased by using more concentrated noise and, lacking consensus, no answer need be given to a student. The consensus answers used are more likely to be correct, offer better intuitive privacy, and incur lower-differential privacy cost. Our evaluation shows our mechanisms improve on the original PATE on all measures, and scale to larger tasks with both high utility and very strong privacy ($\varepsilon < 1.0$).

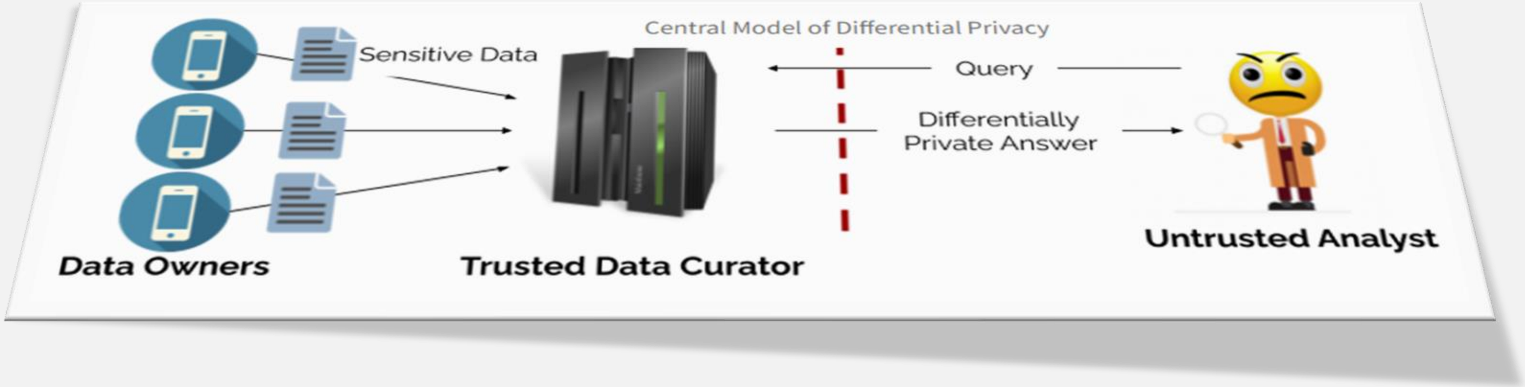
- Written by: Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kuna Talwar, and Úlfar Erlingsson

by Johnny Stuto

# OUTLINE:

- Defenses against Privacy Attacks
- Specific Mitigation Techniques
- Differential Privacy
- Rényi Differential Privacy (RDP)
- PATE Framework Components
- Teachers/Student Model/Aggregator
- Results
- Conclusion
- Questions





# DEFENSES AGAINST PRIVACY ATTACKS

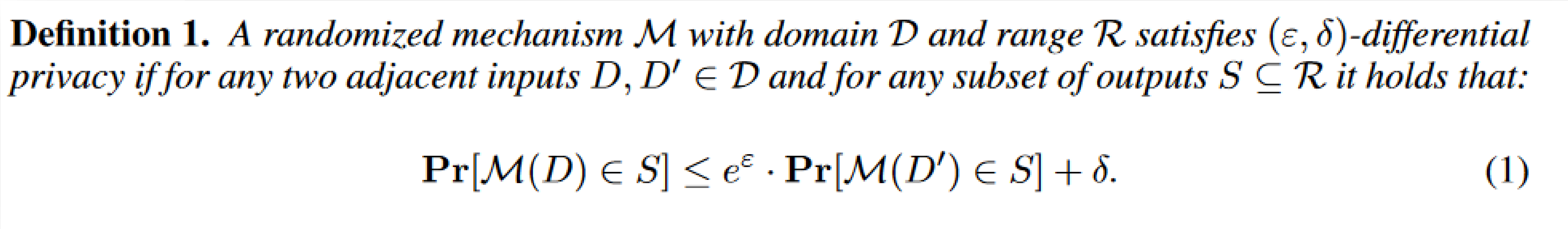
- *Data privacy* techniques have the goal of allowing analysts to learn about *trends* in data, without revealing information specific to *individual data instances*
  - Therefore, privacy techniques involve an intentional release of information, and attempt to control what can be learned from the released information
- Data privacy is rooted in the *Fundamental Law of Information Recovery*, which that, in order to extract accurate and useful information from a dataset, some degree of privacy must be sacrificed.
- Conversely, to achieve strong privacy protection, some loss of utility or accuracy is inevitable
  - This principle highlights the inherent trade-off between the privacy of individual data points and the utility or accuracy of the information derived from the dataset. As stronger privacy guarantees are required, more noise or obfuscation needs to be introduced into the data analysis process, which can result in a decrease in the utility or accuracy of the analysis. There is an inevitable trade-off between privacy and accuracy (i.e., utility)
  - In the context of differential privacy, this trade-off is represented by the privacy parameter (ε). Lower values of ε correspond to stronger privacy guarantees, but at the cost of decreased utility or accuracy in the output. Choosing an appropriate value for ε requires balancing the need for privacy protection with the desired level of utility or accuracy in the analysis.

# MITIGATION TECHNIQUES

- **Anonymization techniques**: Remove personally identifiable information (PII) from the dataset.
- **Data synthesis:** Generate synthetic datasets that preserve the statistical properties of the original data while ensuring privacy.
- **Federated learning**: Use distributed learning approaches, such as federated learning, to train machine learning models on multiple devices or nodes without sharing raw data
- **Secure multi-party computation (SMPC):** Employ cryptographic techniques like SMPC to perform computations on encrypted data without revealing the underlying sensitive information
- **Privacy-aware data collection:** Collect data with privacy considerations in mind from the outset. This includes obtaining proper consent from users, collecting only necessary data, and ensuring proper access controls and data storage practices.
- **Differential privacy:** Implement differentially private algorithms for data analysis and machine learning. This approach introduces controlled noise to the output of an algorithm, providing a mathematically provable privacy guarantee. Balancing the privacy parameter (ε) with the desired utility or accuracy is crucial to ensure a reasonable trade-off between privacy and utility.

# DIFFERENTIAL PRIVACY:

- The PATE (Private Aggregation of Teacher Ensembles) method falls under the category of differential privacy.
- PATE is a privacy-preserving machine learning framework that enables model-agnostic training while providing differential privacy guarantees for the training dataset.
- By using the PATE method, you can train machine learning models that preserve privacy without significantly compromising performance, making it a valuable approach for various applications where data privacy is a primary concern.
  - Injecting noise into the dataset to create plausible deniability.
  - The noise shouldn't change the outcome of the computation.
  - An observer should not be able to determine any PII from the output or identify whose data was trained on.



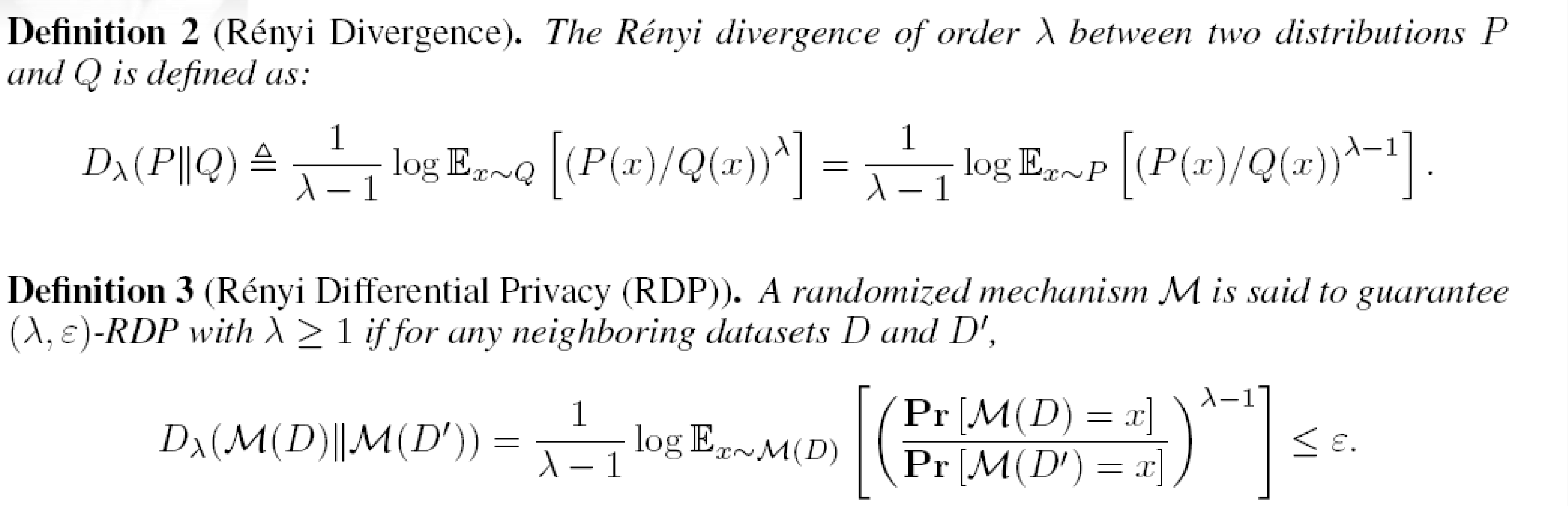
**Definition 1.** *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies $(\varepsilon, \delta)$-differential privacy if for any two adjacent inputs $D, D' \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that:*

$$\mathbf{Pr}[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \mathbf{Pr}[\mathcal{M}(D') \in S] + \delta. \tag{1}$$

ε = Upper bound for the loss of privacy
δ = probability that privacy will not be held
M = Model training algorithm

# RÉNYI DIFFERENTIAL PRIVACY

- Rényi divergence, also known as Rényi's α-divergence, is a measure of dissimilarity between two probability distributions. It was introduced by Alfréd Rényi, a Hungarian mathematician, in 1961. Rényi divergence generalizes the concept of the Kullback-Leibler (KL) divergence, a widely used measure of divergence between probability distributions.
- Rényi Differential Privacy (RDP) is a generalization of the classical differential privacy framework. It was introduced by Ilya Mironov in 2017 to address some limitations of traditional differential privacy, particularly in the context of composition, which occurs when multiple privacy-preserving operations or queries are applied. RDP introduces an additional parameter, called the order (α), alongside the classical privacy parameter (ε).



**Definition 2** (Rényi Divergence). *The Rényi divergence of order $\lambda$ between two distributions $P$ and $Q$ is defined as:*

$$D_\lambda(P\|Q) \triangleq \frac{1}{\lambda-1}\log \mathbb{E}_{x\sim Q}\left[\left(P(x)/Q(x)\right)^{\lambda}\right] = \frac{1}{\lambda-1}\log \mathbb{E}_{x\sim P}\left[\left(P(x)/Q(x)\right)^{\lambda-1}\right].$$

**Definition 3** (Rényi Differential Privacy (RDP)). *A randomized mechanism $\mathcal{M}$ is said to guarantee $(\lambda,\varepsilon)$-RDP with $\lambda \geq 1$ if for any neighboring datasets $D$ and $D'$,*

$$D_\lambda(\mathcal{M}(D)\|\mathcal{M}(D')) = \frac{1}{\lambda-1}\log \mathbb{E}_{x\sim\mathcal{M}(D)}\left[\left(\frac{\mathbf{Pr}\left[\mathcal{M}(D)=x\right]}{\mathbf{Pr}\left[\mathcal{M}(D')=x\right]}\right)^{\lambda-1}\right] \leq \varepsilon.$$

# PATE FRAMEWORK COMPONENTS

- **Current approaches show potential but are untested at scale and have issues with scalability, robustness and utility.**
- In the PATE framework, multiple teacher models are trained on disjoint subsets of the sensitive training dataset
- A student model then learns from these teacher models by querying them through an aggregator, which enforces differential privacy by introducing controlled noise.
- The student model is trained on insensitive data, allowing it to generalize the knowledge acquired from the teacher models without directly accessing sensitive data.

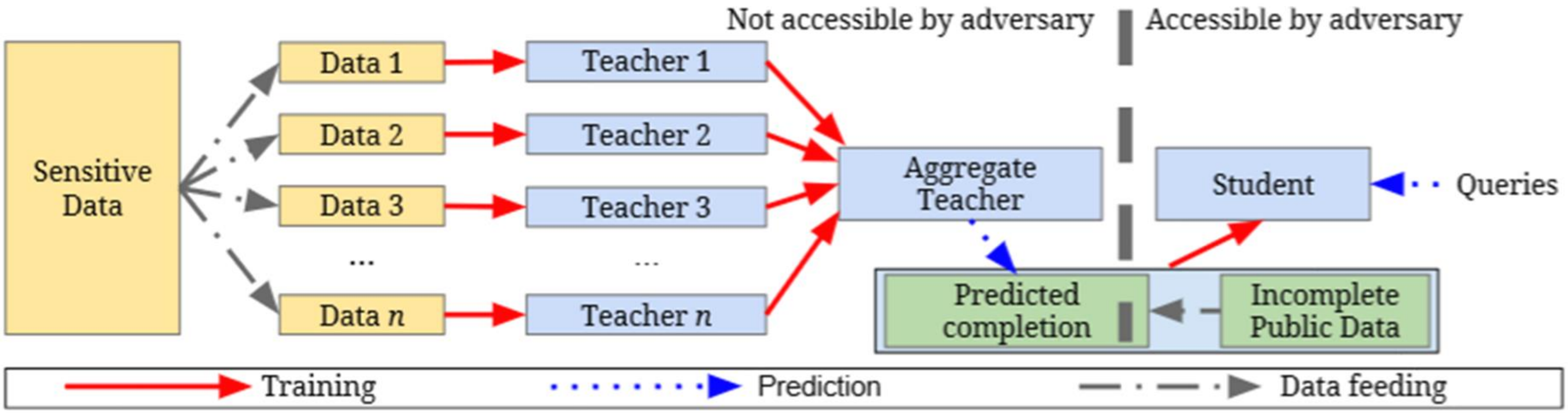
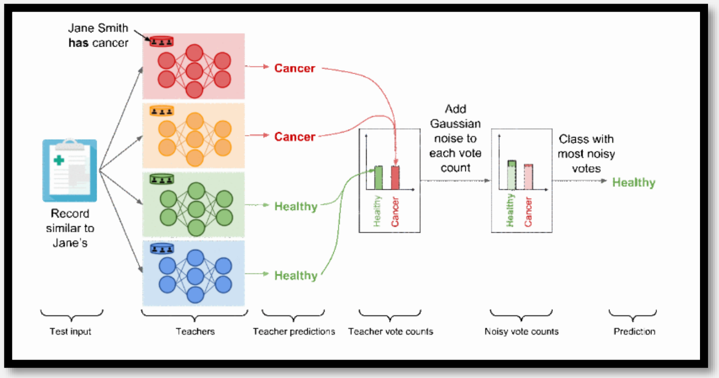


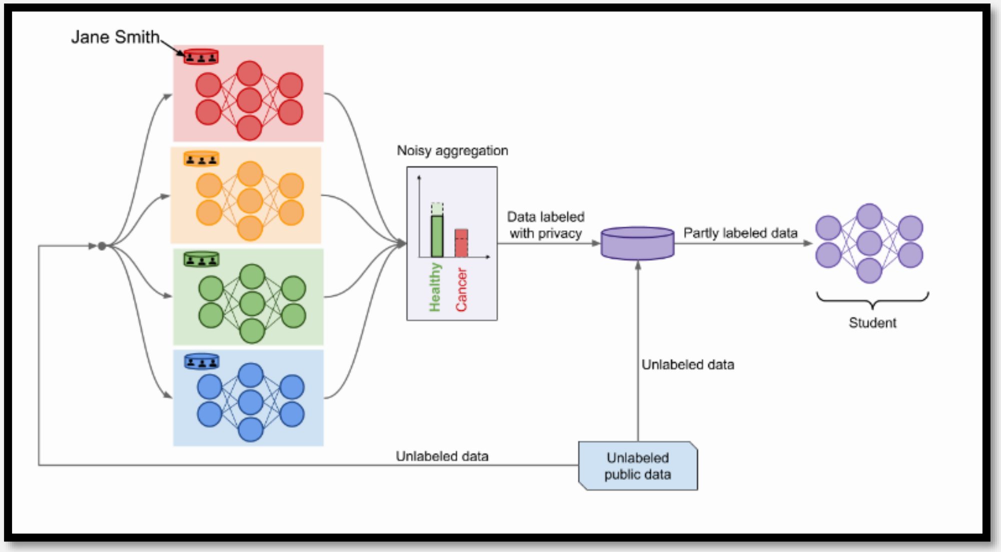
Figure 2: Overview of the approach: (1) an ensemble of teachers is trained on disjoint subsets of the sensitive data, (2) a student model is trained on public data labeled using the ensemble.

# PATE FRAMEWORK



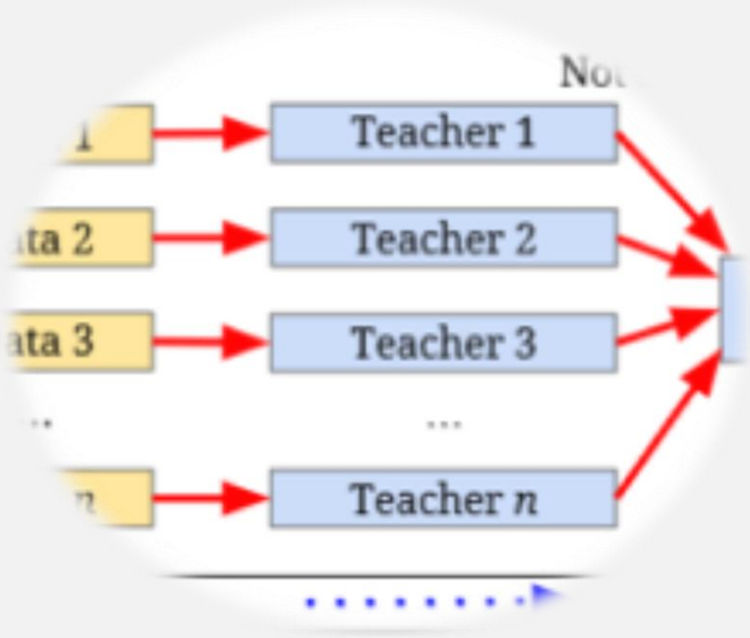
The **PATE approach** is based on a simple intuition: if two different classifiers, trained on two different datasets with no training examples in common, agree on how to classify a new input example, then that decision does not reveal information about any single training example.

The decision could have been made with or without any single training example, because both the model trained with that example and the model trained without that example reached the same conclusion.



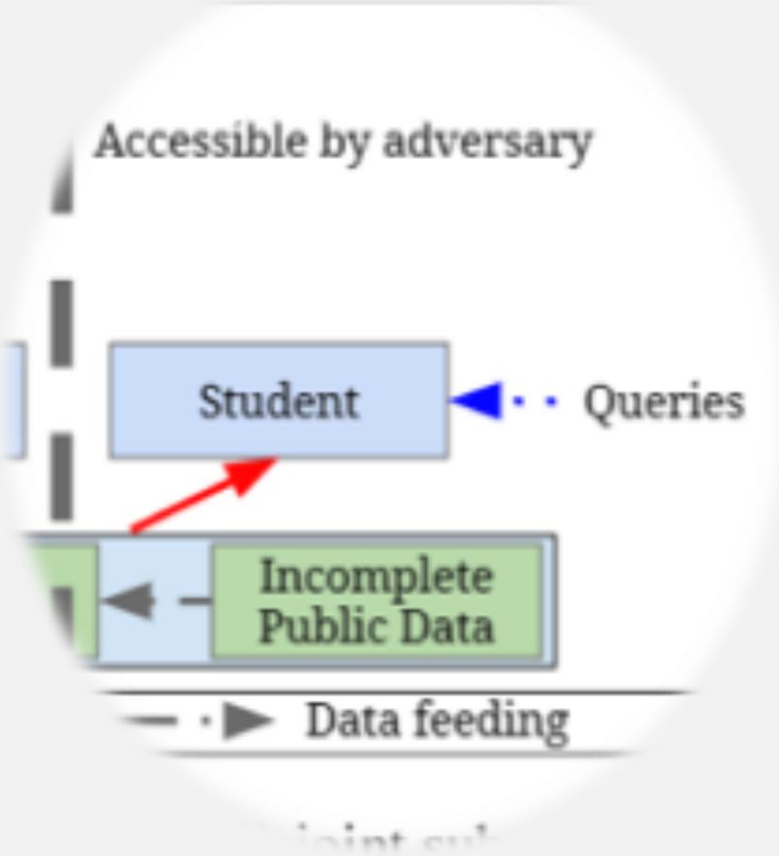
# TEACHERS

- The teacher models are trained on disjoint subsets of the sensitive training dataset, each with its own portion of the data.
- These models provide expertise on the data they were trained on while maintaining privacy.
- By using multiple teachers, the PATE framework leverages the wisdom of the ensemble, which helps increase the overall accuracy and generalization capabilities of the student model.

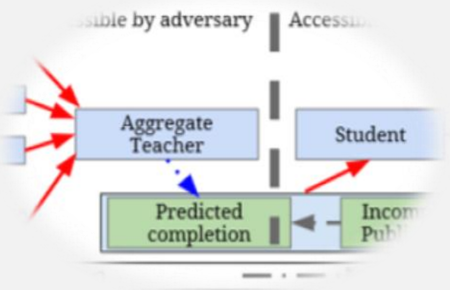


# STUDENT MODELS

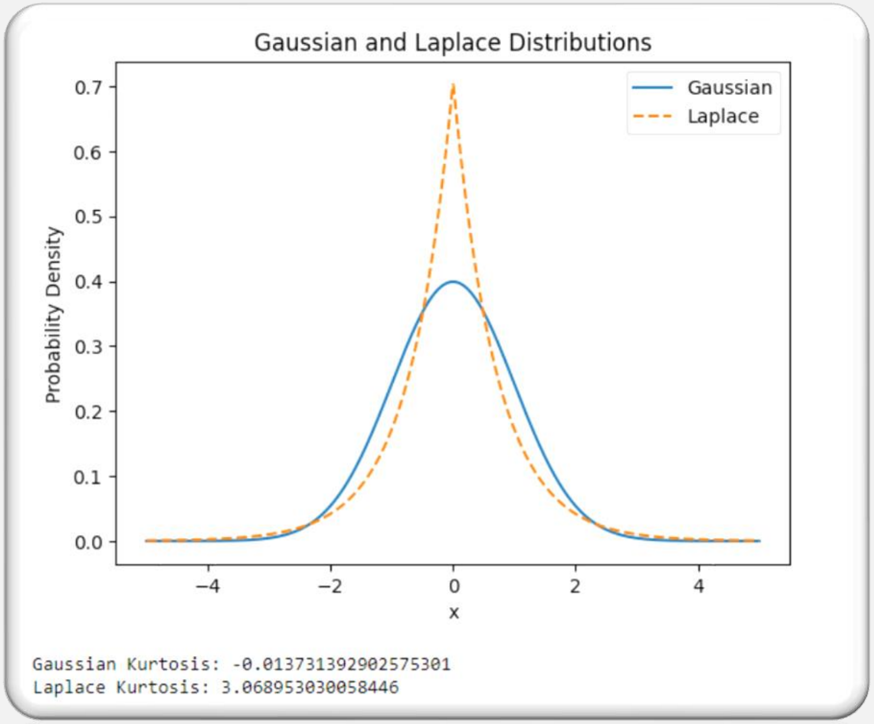
- The student model learns from the teacher models without direct access to the sensitive data. It is trained on an insensitive or public unlabeled dataset, which is labeled by interacting with the ensemble of teachers via the aggregator.
- The student model aims to replicate the performance of the teachers while ensuring privacy protection for the sensitive training data



# AGGREGATORS



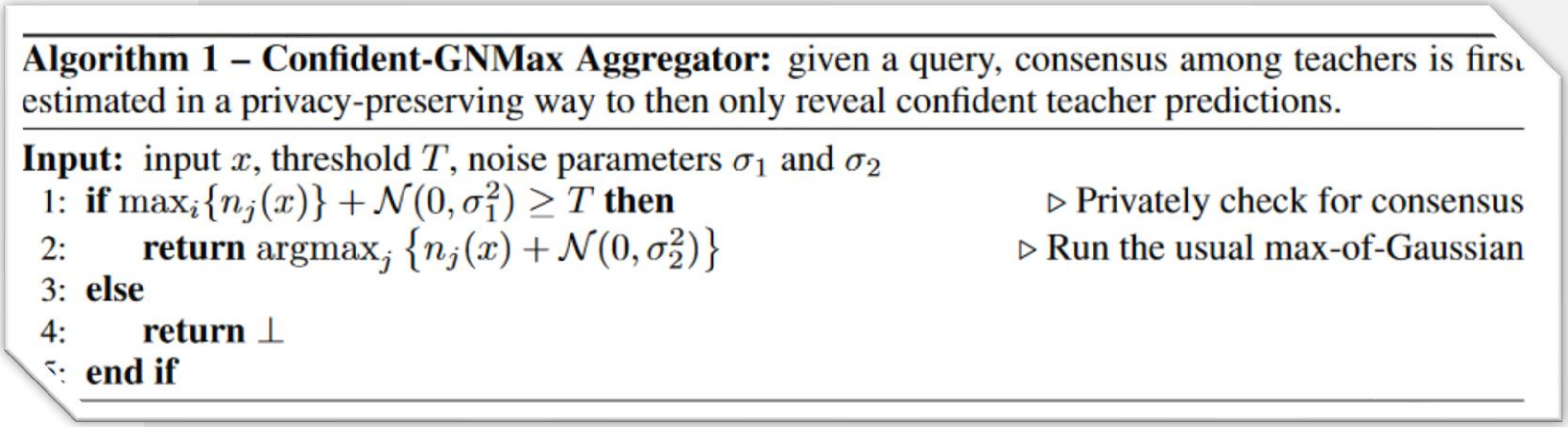
- Build off the PATE method
  - New methods for aggregating teacher/student answers
    - Confidence Aggregator
      - Teacher consensus module where Min of T teachers guarantee a correct classification and throw out queries where teachers don't know
    - Interactive Aggregator
      - Student confidence scoring. Don't ask teachers for an answer if confidence is high that the student knows
    - Expensive Queries are high cost to privacy
  - Gaussian noise instead of Laplacian
    - Less computationally expensive
    - Causes less noise overall.



```
Gaussian Kurtosis: -0.013731392902575301
Laplace Kurtosis: 3.068953030058446
```

# AGGREGATOR ALGORITHMS

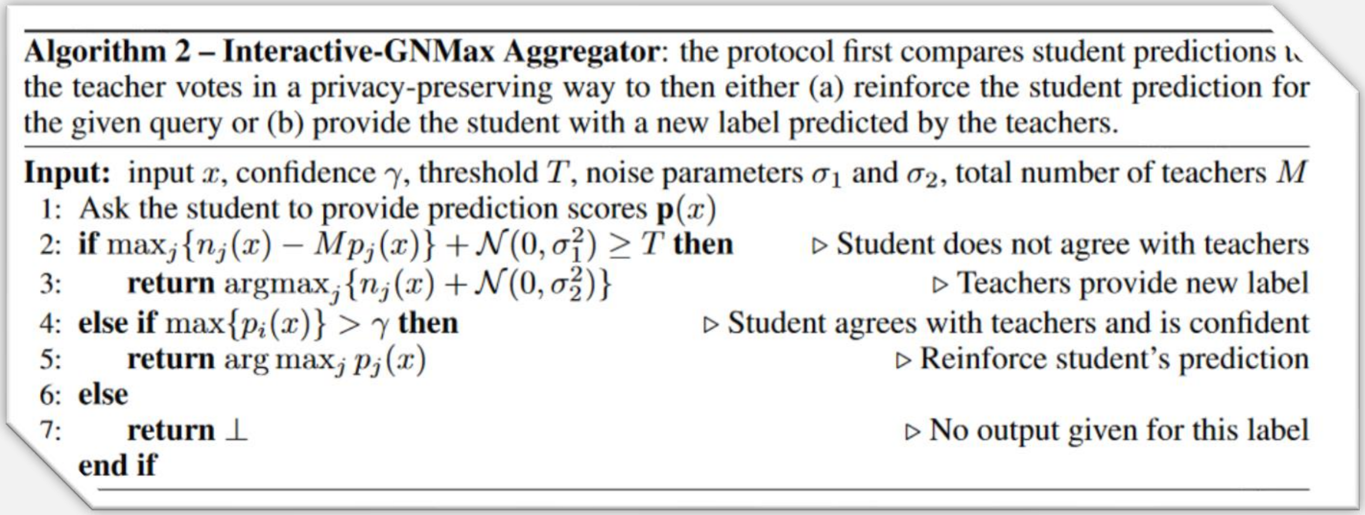
## Confident Aggregator:



**Algorithm 1 – Confident-GNMax Aggregator:** given a query, consensus among teachers is first estimated in a privacy-preserving way to then only reveal confident teacher predictions.

```
Input: input $x$, threshold $T$, noise parameters $\sigma_1$ and $\sigma_2$
if $\max_i\{n_j(x)\} + \mathcal{N}(0, \sigma_1^2) \geq T$ then        ▷ Privately check for consensus
    return $\arg\max_j \left\{n_j(x) + \mathcal{N}(0, \sigma_2^2)\right\}$        ▷ Run the usual max-of-Gaussian
else
    return $\perp$
end if
```

## Interactive Aggregator:



**Algorithm 2 – Interactive-GNMax Aggregator**: the protocol first compares student predictions to the teacher votes in a privacy-preserving way to then either (a) reinforce the student prediction for the given query or (b) provide the student with a new label predicted by the teachers.

```
Input: input $x$, confidence $\gamma$, threshold $T$, noise parameters $\sigma_1$ and $\sigma_2$, total number of teachers $M$
Ask the student to provide prediction scores $\mathbf{p}(x)$
if $\max_j\{n_j(x) - Mp_j(x)\} + \mathcal{N}(0, \sigma_1^2) \geq T$ then        ▷ Student does not agree with teachers
    return $\arg\max_j\{n_j(x) + \mathcal{N}(0, \sigma_2^2)\}$        ▷ Teachers provide new label
else if $\max\{p_i(x)\} > \gamma$ then        ▷ Student agrees with teachers and is confident
    return $\arg\max_j p_j(x)$        ▷ Reinforce student's prediction
else
    return $\perp$        ▷ No output given for this label
end if
```

# EXPERIMENTAL SETUP

- Datasets (private personal attributes)
  - MNIST
  - Street View House Numbers (SVHN)
  - US Census Income Adult (UCI Adult
  - Glyph: Synthetically generated computer font symbols with at most 150 different classes
- Teacher Ensembles: 100,500,1000,5000 (number of teachers & partitions of data)
- Queries: 500-12000 depending on dataset
- Privacy parameters: $\delta = 10^{-8}$ probability that privacy will not be held

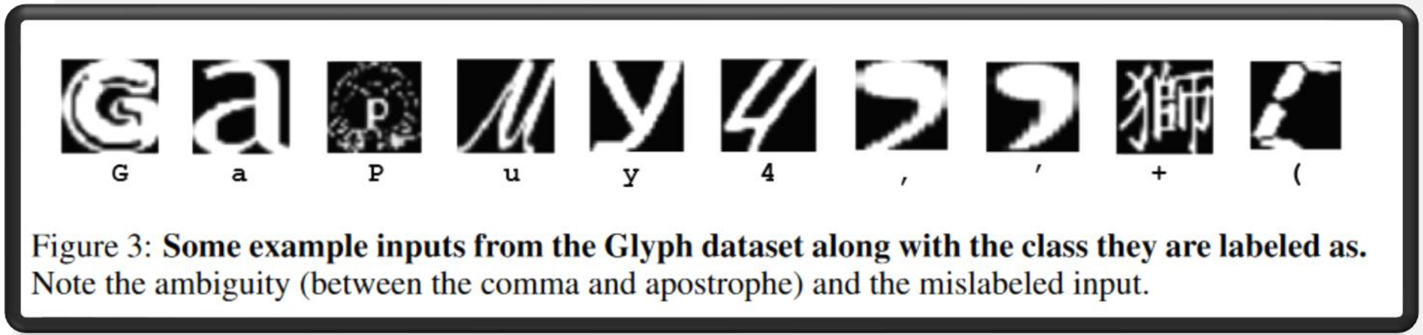
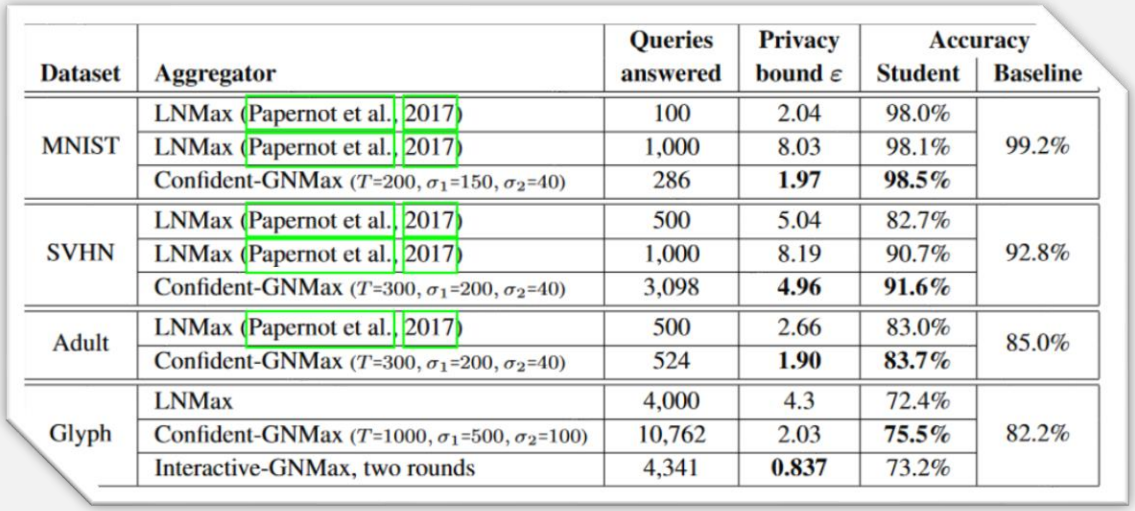


Figure 3: **Some example inputs from the Glyph dataset along with the class they are labeled as.** Note the ambiguity (between the comma and apostrophe) and the mislabeled input.

# RESULTS



| Dataset | Aggregator | Queries answered | Privacy bound $\varepsilon$ | Accuracy Student | Accuracy Baseline |
|---|---|---|---|---|---|
| MNIST | LNMax (Papernot et al., 2017) | 100 | 2.04 | 98.0% | 99.2% |
|  | LNMax (Papernot et al., 2017) | 1,000 | 8.03 | 98.1% |  |
|  | Confident-GNMax ($T$=200, $\sigma_1$=150, $\sigma_2$=40) | 286 | **1.97** | **98.5%** |  |
| SVHN | LNMax (Papernot et al., 2017) | 500 | 5.04 | 82.7% | 92.8% |
|  | LNMax (Papernot et al., 2017) | 1,000 | 8.19 | 90.7% |  |
|  | Confident-GNMax ($T$=300, $\sigma_1$=200, $\sigma_2$=40) | 3,098 | **4.96** | **91.6%** |  |
| Adult | LNMax (Papernot et al., 2017) | 500 | 2.66 | 83.0% | 85.0% |
|  | Confident-GNMax ($T$=300, $\sigma_1$=200, $\sigma_2$=40) | 524 | **1.90** | **83.7%** |  |
| Glyph | LNMax | 4,000 | 4.3 | 72.4% | 82.2% |
|  | Confident-GNMax ($T$=1000, $\sigma_1$=500, $\sigma_2$=100) | 10,762 | 2.03 | **75.5%** |  |
|  | Interactive-GNMax, two rounds | 4,341 | **0.837** | 73.2% |  |

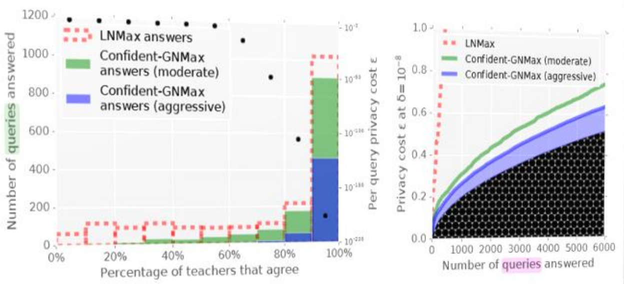


Figure 5: **Effects of the noisy threshold checking:** *Left:* The number of queries answered by LNMax, Confident-GNMax moderate ($T$=3500, $\sigma_1$=1500), and Confident-GNMax aggressive ($T$=5000, $\sigma_1$=1500). The black dots and the right axis (in log scale) show the expected cost of answering a single query in each bin (via GNMax, $\sigma_2$=100). *Right:* Privacy cost of answering all (LNMax) vs only inexpensive queries (GNMax) for a given number of answered queries. The very dark area under the curve is the cost of selecting queries; the rest is the cost of answering them.

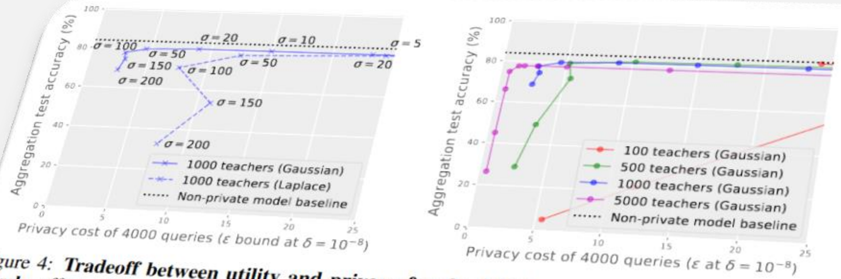


Figure 4: **Tradeoff between utility and privacy for the LNMax and GNMax aggregators on Glyph:** effect of the noise distribution (left) and size of the teacher ensemble (right). The LNMax aggregator uses a Laplace distribution and GNMax a Gaussian. Smaller values of the privacy cost $\varepsilon$ (often obtained by increasing the noise scale $\sigma$—see Section 4) and higher accuracy are better.

# CONCLUSION

- Privacy can be thought of as an ally rather than a foe in the context of machine learning.
- As the techniques improve, differential privacy is likely to serve as an effective regularizer that produces better-behaved models.
- Within the framework of PATE, machine learning researchers can also make significant contributions towards improving differential privacy guarantees without being an expert in the formal analyses behind these guarantees.
- New methods for aggregating teacher/student answers provide a privacy preserving technique that reduces leakage
  - Caps queries at a confidence interval
  - Stops overfitting student’s queries by checking confidence of student's answers
- Improvements across the board in Privacy ε  loss(lower is better)
- Confirms that PATE has the potential to be used at scale
  - Generalization improved by changing perturbation method

# REFERENCES

- Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú., ***Scalable Private Learning with PATE,*** arXiv e-prints, 2018.
- Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. ***Semi supervised knowledge transfer for deep learning from private training data***. In Proceedings ofthe 5th International Conference on Learning Representations (ICLR), 2017.
- Liu et al. (2020) When Machine Learning Meets Privacy: A Survey and Outlook
- Rigaki and Carcia (2021) A Survey of Privacy Attacks in Machine Learning  
  Cristofaro (2020) An Overview of Privacy in Machine Learning
- https://www.nist.gov/blogs/cybersecurity-insights/threat-models-differential-privacy
- http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

# Introduction to Federated Learning

Presenter: Shahidur Rahoman Sohag

Reporting To:

Dr. Alex Vakanski

# Table of Contents

- ❑ **Introduction**

  Concept & Motivation (What & Why)

- ❑ **Mechanism**

  How it works, popular optimization algorithms

- ❑ **Research Findings**

  Different research methods, pros, cons

- ❑ **Conclusion**

  Future research, expectations, my thoughts

# DID YOU KNOW?

According to the U.S. Department of Health and Human Services, the 337 healthcare incidents in 2022 **reported affected 19,992,810 individuals.**

# Introduction

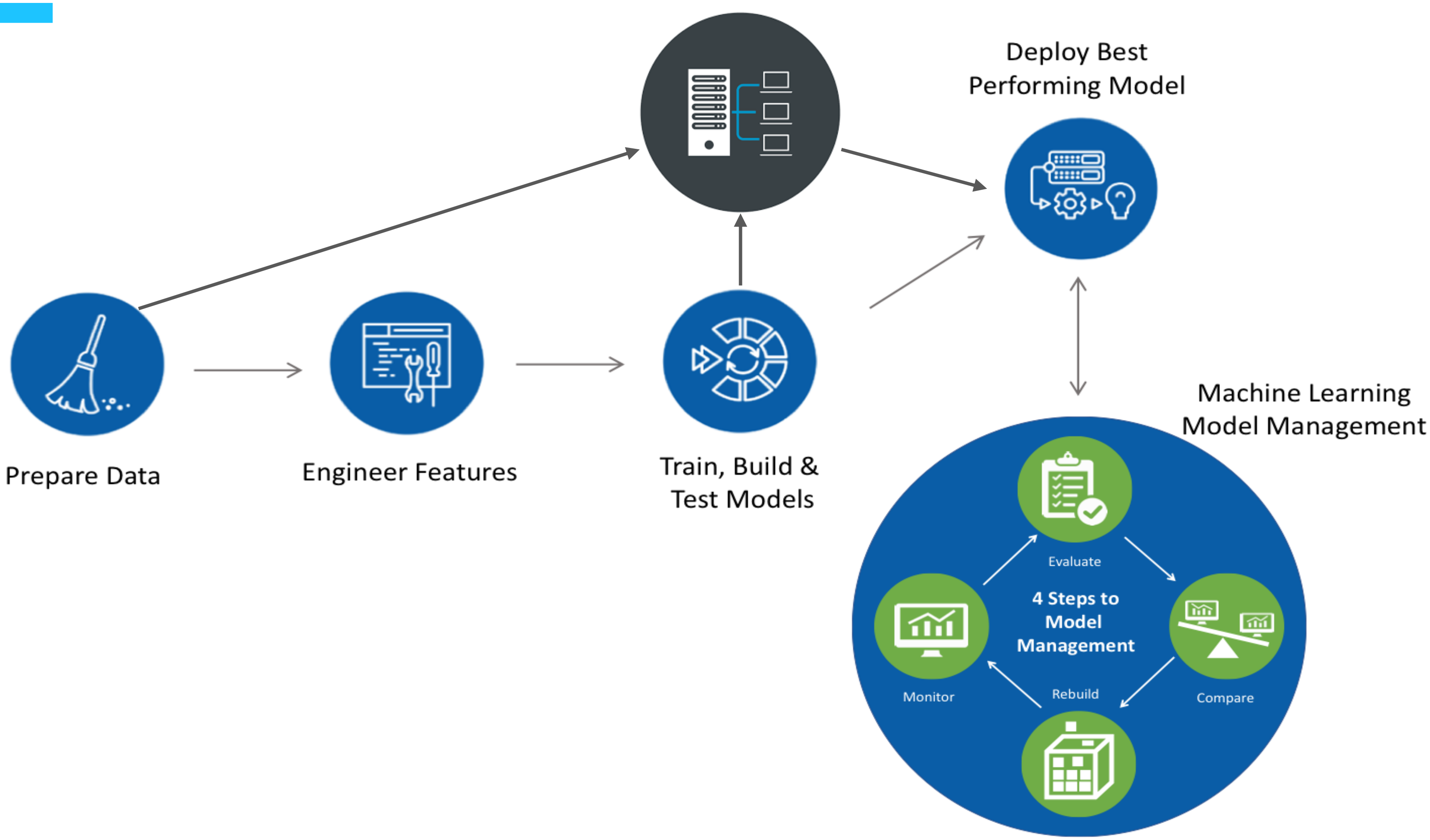
- **What is federated learning?**

  Federated learning (also known as collaborative learning) is a machine learning technique that trains an algorithm via multiple independent sessions, each using its own dataset

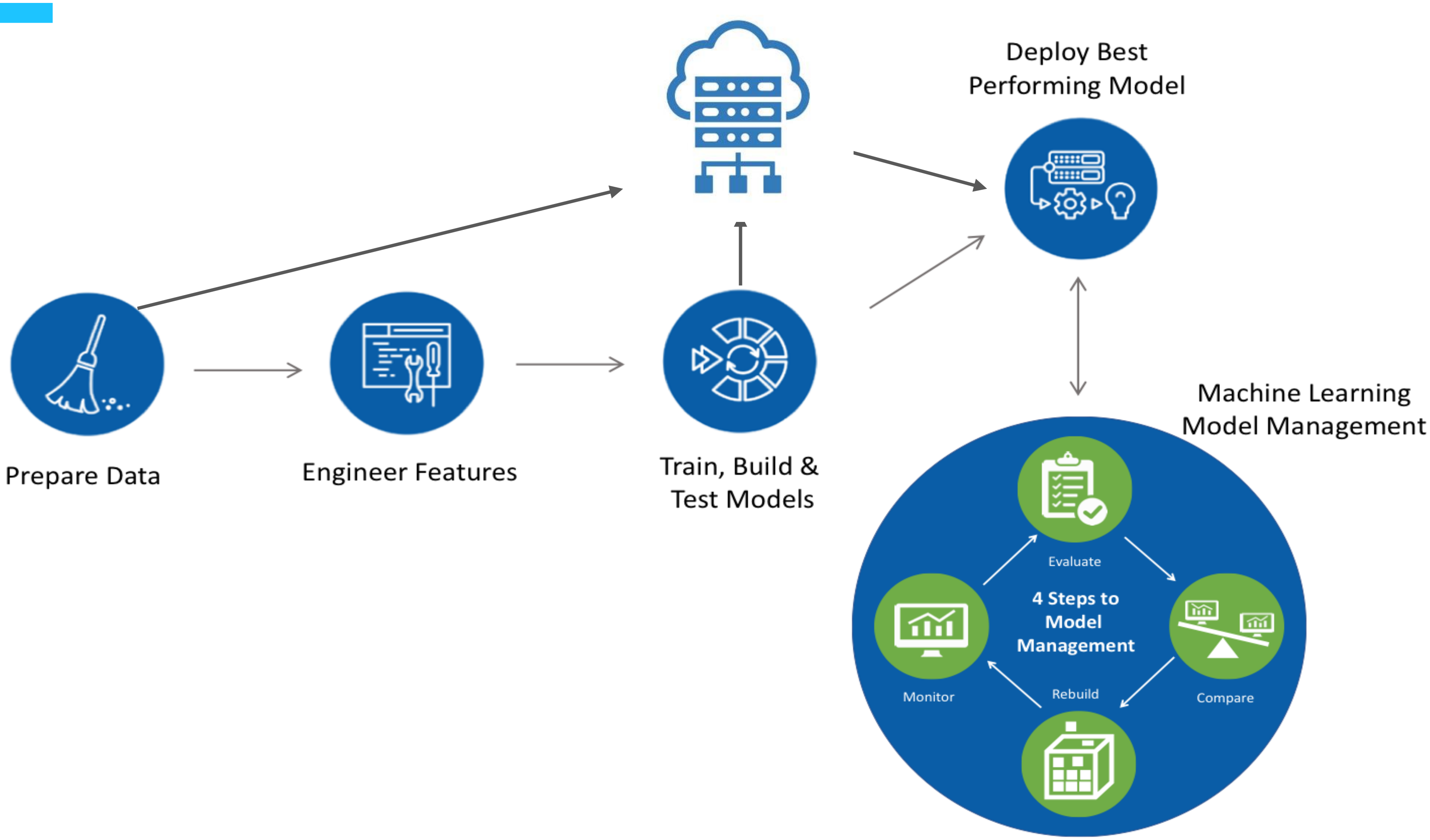
- **Motivation?**

  The motivation for federated learning is the preservation of the privacy of the data owned by the clients.

# General ML Approach



# Cloud ML Approach



Deploy Best Performing Model

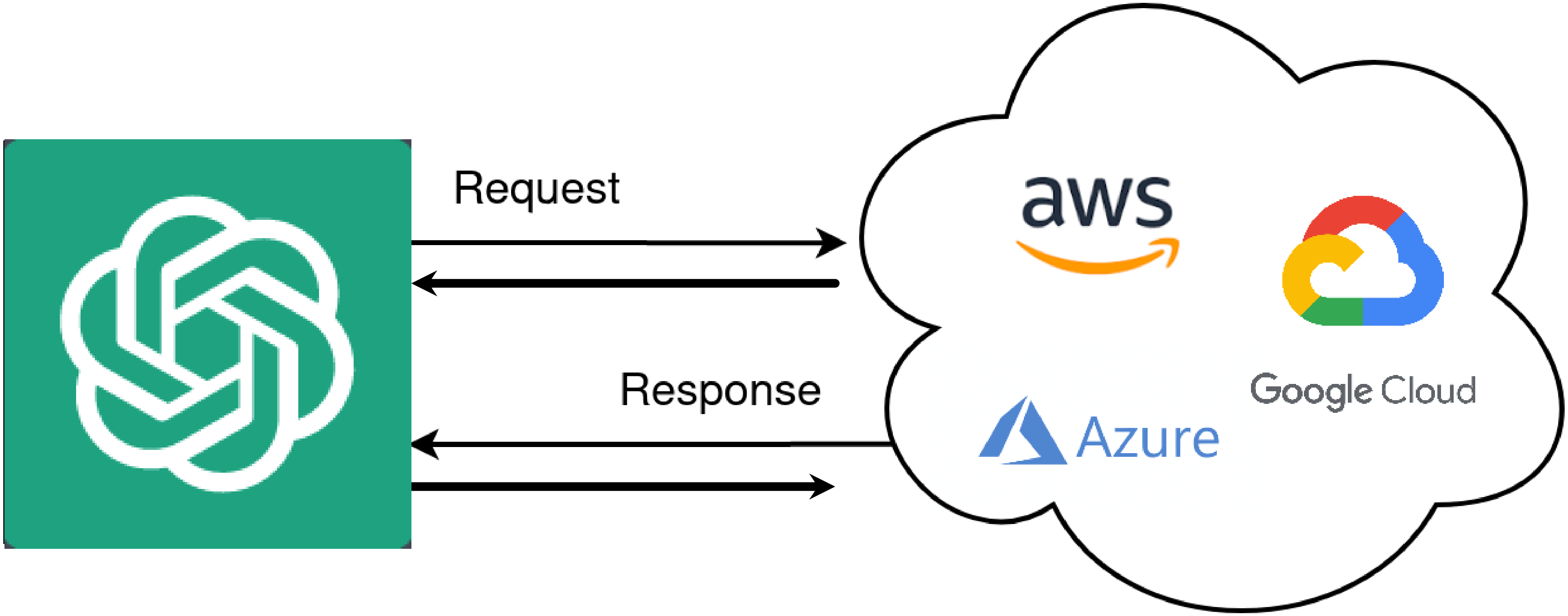
Prepare Data

Engineer Features

Train, Build & Test Models

Machine Learning Model Management

# Centralized ML Approach



# WHAT'S WRONG WITH THIS?

# Drawbacks of Centralized ML

## Privacy Concerns

CML involves transmitting all the data to a central location.

## Scalability Issues

Difficult to scale as the number of users and data sets increase

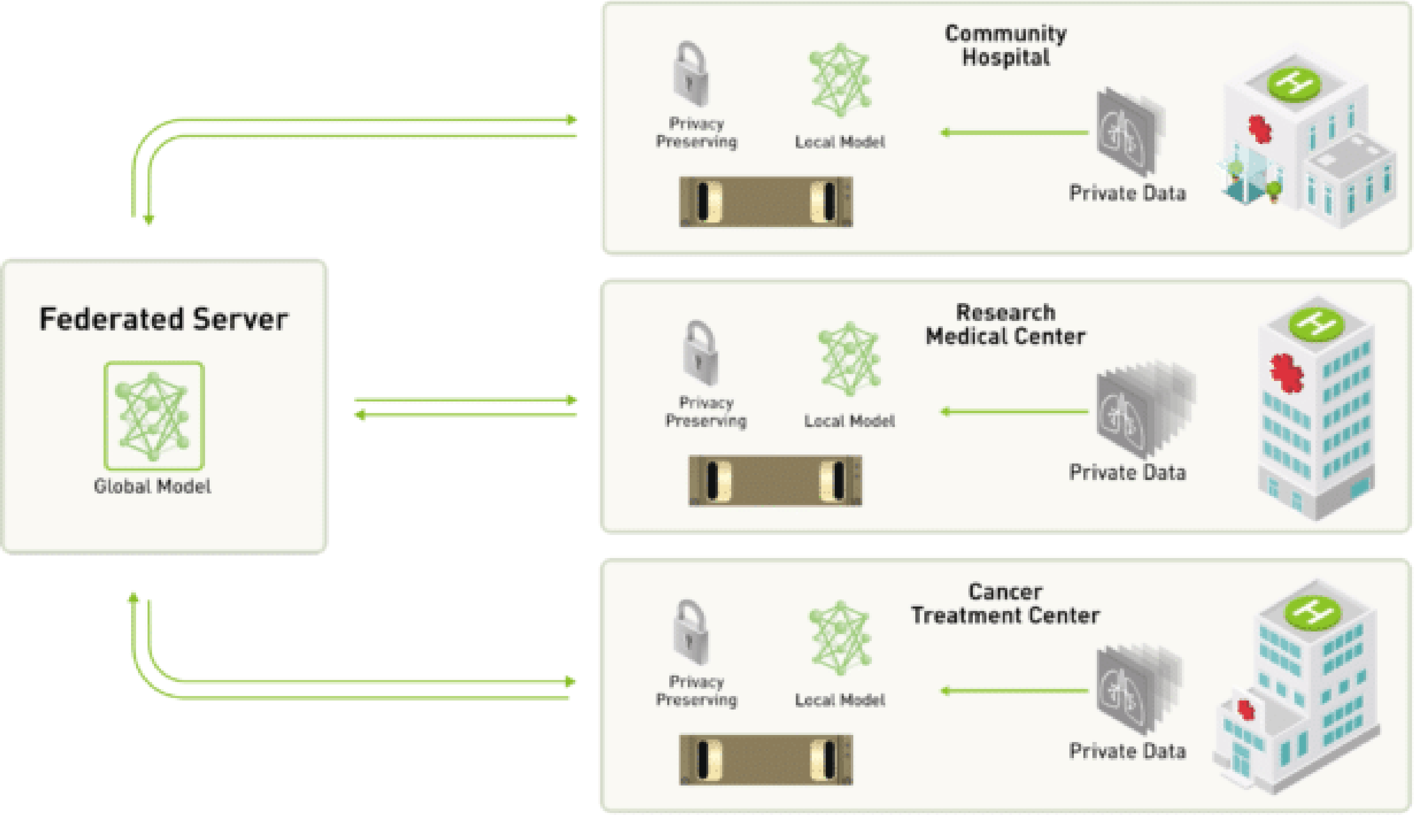
## Network Latency

Big issue when the data being transmitted is large to central server for processing

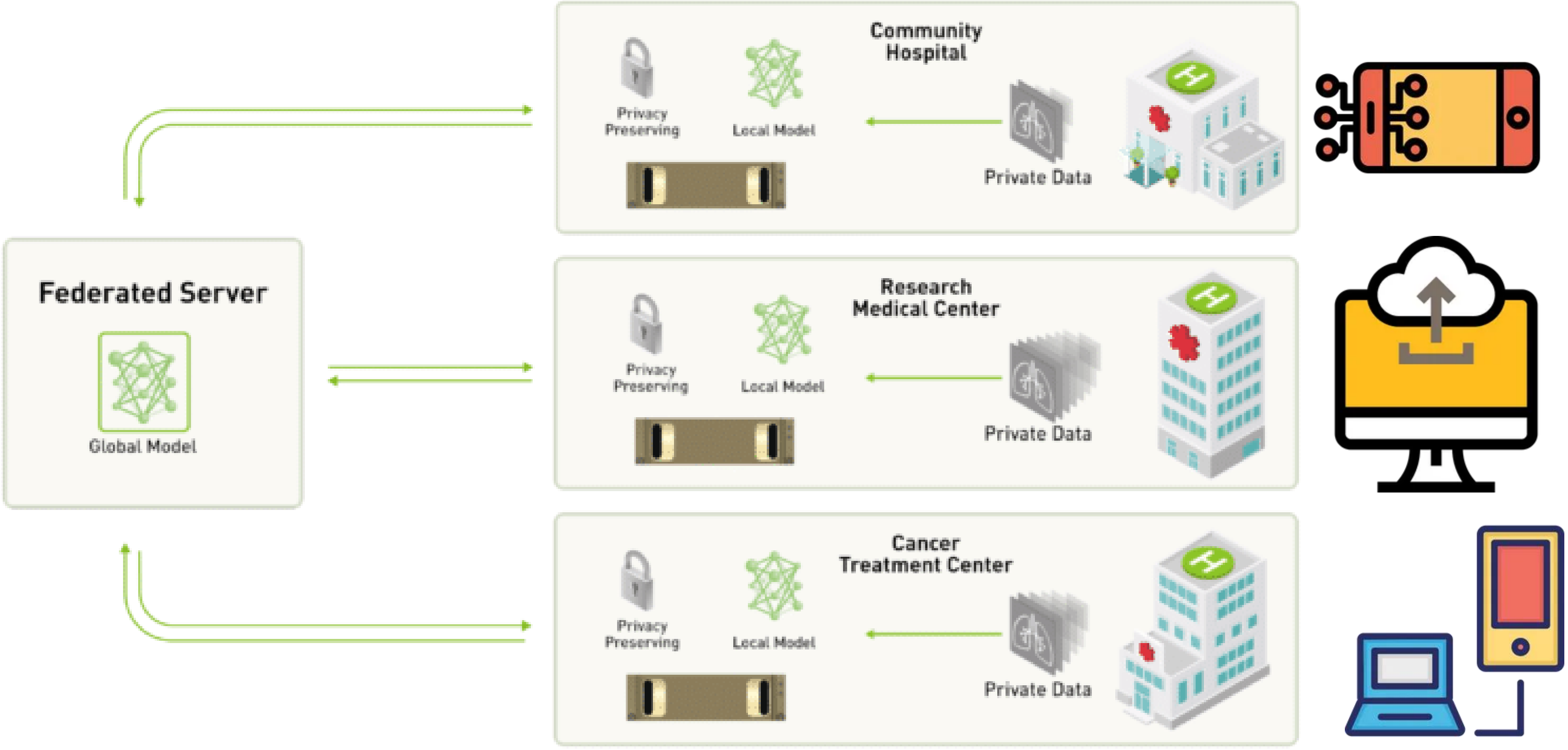
## Cost & Bias

Requires significant computing resources. Can be prone to bias if the training data is not representative of the population,

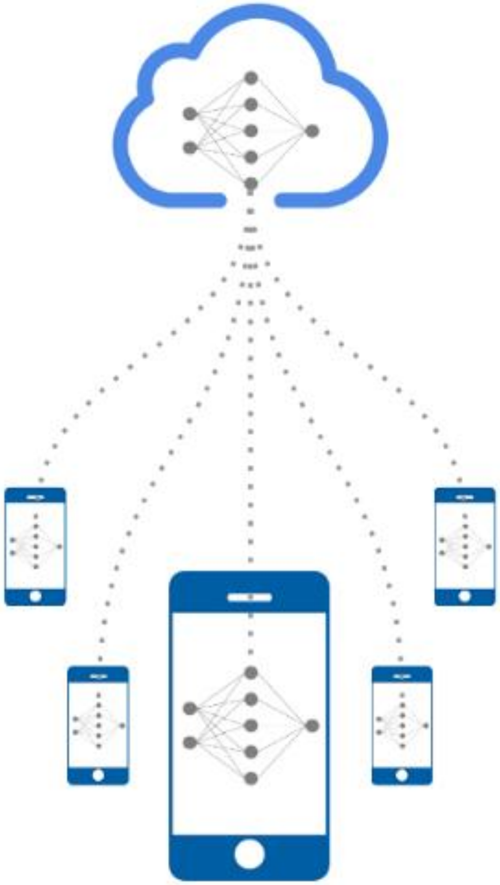
# Federated Learning



# Federated Learning (Heterogenous)



# Key Factors of Federated Learning



Hyper-Personalized

Low Cloud Infra Overheads

Minimum Latencies

Privacy Preserving

# Optimization Algorithm

## FedSGD

This corresponds to a full-batch (non-stochastic) gradient descent. For the current global model *w*$^t$, the average gradient on its global model is calculated for each client *k*.

$w_t$ - Model weights on communication round $\#t$

$w_t^k$ - Model weights on communication round $\#t$ on client $k$

$C$ - Fraction of clients performing computations in each round

$E$ - Number of training passes each client makes over its local dataset on each round

$B$ - The local minibatch size used for the client updates

$\eta$ - The learning rate

$\mathcal{P}_k$ - Set of data points on client $k$

$n_k$ - Number of data points on client $k$

$f_i(w)$ - Loss $l(x_i, y_i; w)$ i.e., loss on example $(x_i, y_i)$ with model parameters $w$

$$F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w)$$

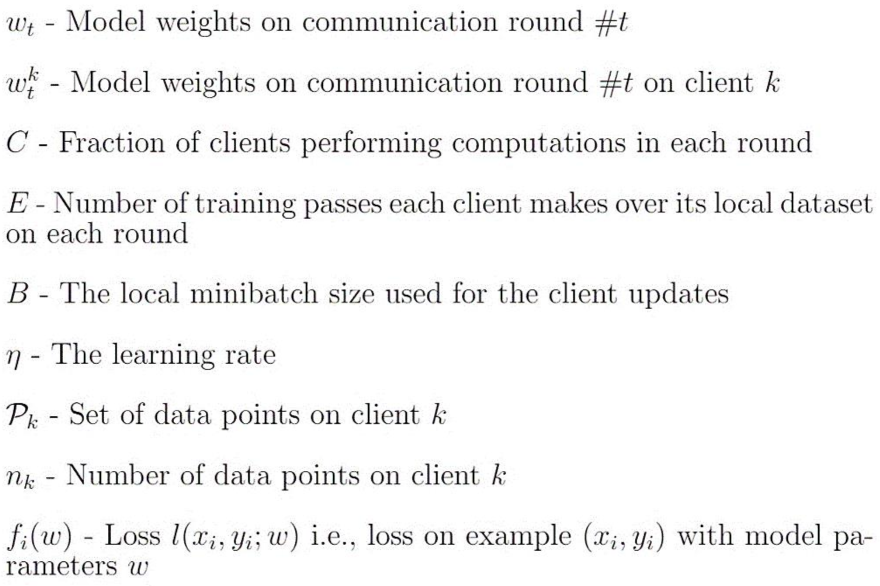
$$g_k = \nabla F_k(w_t)$$

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

# Optimization Algorithm

## FedAVG

Each client locally takes one step of gradient descent on the current model using its local data, and the server then takes a weighted average of the resulting models.



$w_t$ - Model weights on communication round $\#t$

$w_t^k$ - Model weights on communication round $\#t$ on client $k$

$C$ - Fraction of clients performing computations in each round

$E$ - Number of training passes each client makes over its local dataset on each round

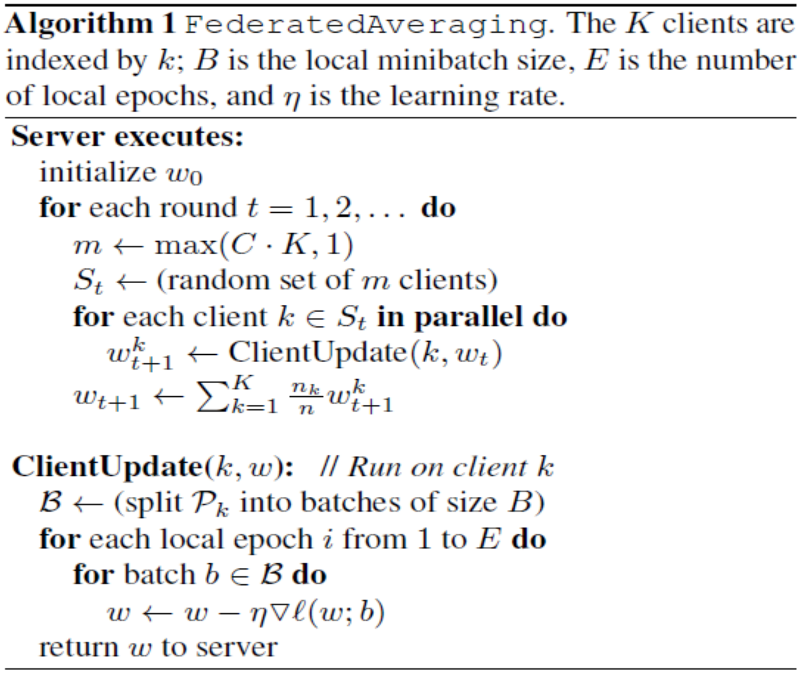
$B$ - The local minibatch size used for the client updates

$\eta$ - The learning rate

$\mathcal{P}_k$ - Set of data points on client $k$

$n_k$ - Number of data points on client $k$

$f_i(w)$ - Loss $l(x_i, y_i; w)$ i.e., loss on example $(x_i, y_i)$ with model parameters $w$

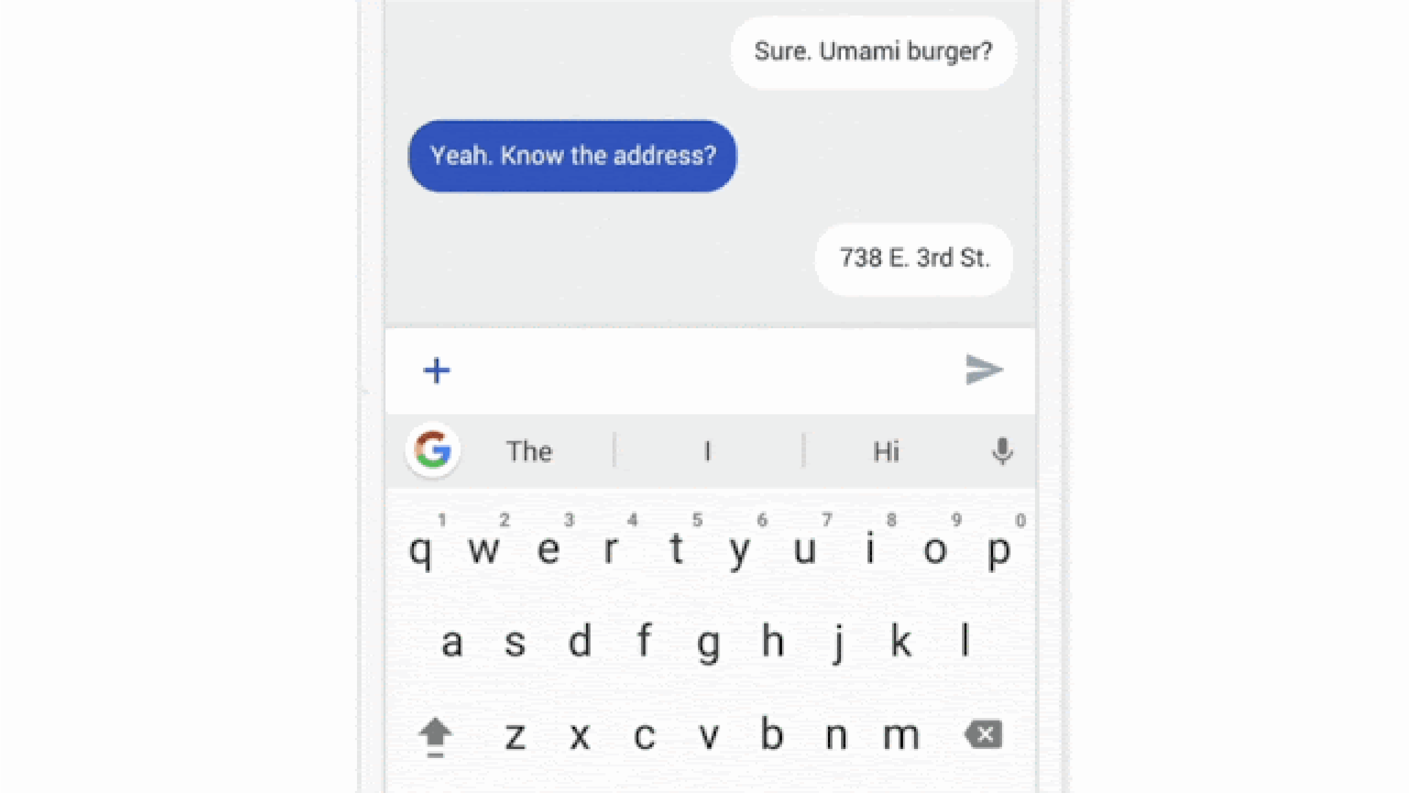


**Algorithm 1** `FederatedAveraging`. The $K$ clients are indexed by $k$; $B$ is the local minibatch size, $E$ is the number of local epochs, and $\eta$ is the learning rate.

```
Server executes:
  initialize $w_0$
  for each round $t = 1, 2, \dots$ do
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of $m$ clients)
    for each client $k \in S_t$ in parallel do
      $w_{t+1}^k \leftarrow$ ClientUpdate$(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate$(k, w)$:  // Run on client $k$
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size $B$)
  for each local epoch $i$ from 1 to $E$ do
    for batch $b \in \mathcal{B}$ do
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return $w$ to server
```

# A Regular Application of FL



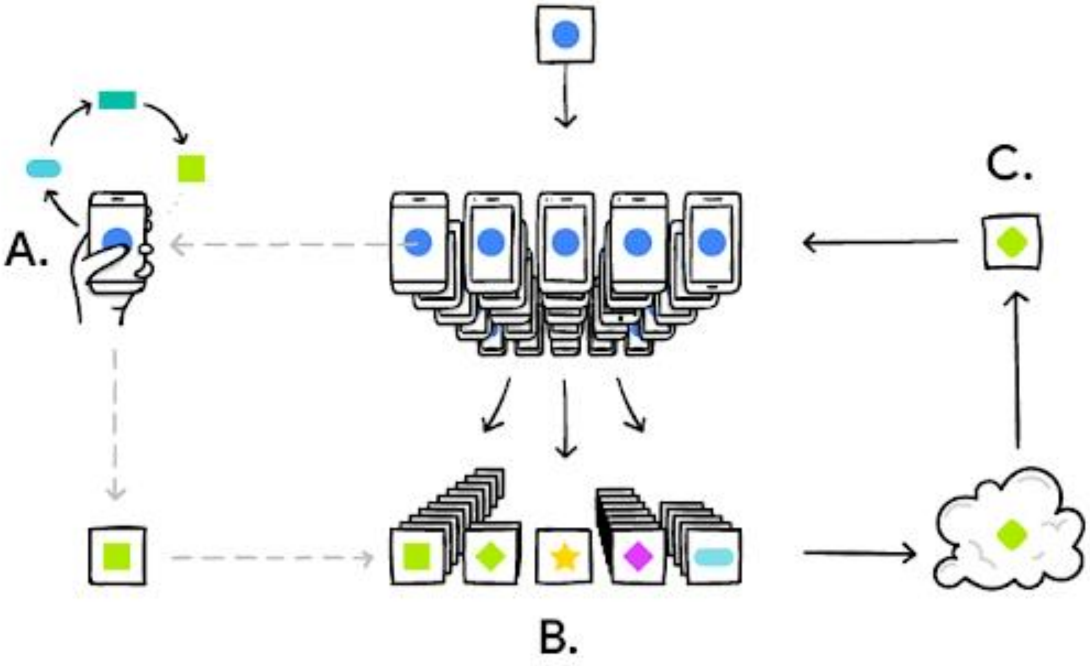
Sure. Umami burger?

Yeah. Know the address?

738 E. 3rd St.

# Behind The Scene

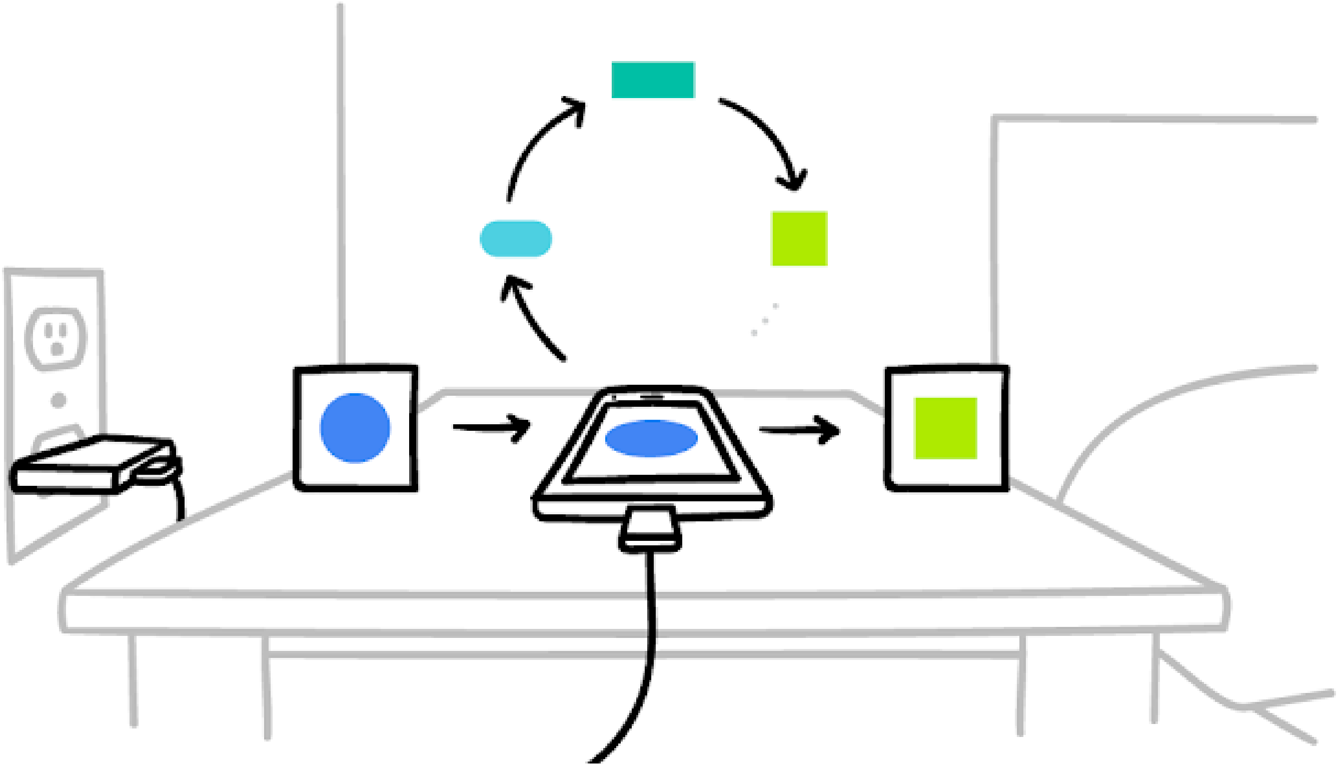
Personalizes model locally based on users usage



Sends the updated result, and repeats

Many users updates are aggregated to form a concensus change

# When FL Updates?

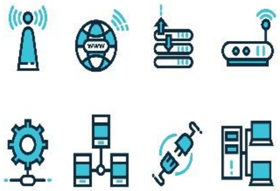


# Current Research

## Privacy-preserving federated learning

One of the main challenges in federated learning is to ensure the privacy of the local data on the devices. Researchers are exploring new methods to improve the privacy of federated learning algorithms, such as using differential privacy or homomorphic encryption

## Communication-efficient federated learning



Federated learning involves communication between the devices and the central server, which can be a bottleneck in terms of time and resources. Researchers are exploring new methods to reduce the communication overhead of federated learning algorithms, such as using compression techniques or designing more efficient communication protocols.

# Current Research

## Federated learning in non-i.i.d. settings

Federated learning assumes that the data on the devices are identically and independently distributed (i.i However, in real-world scenarios, this assumption may not hold. Researchers are exploring new methods extend federated learning to non-i.i.d. settings, such as using transfer learning or meta-learning.

## Federated learning for healthcare

Federated learning has the potential to improve healthcare by enabling the development of predictive models without compromising patient privacy. Researchers are exploring new methods to apply federated learning to healthcare applications, such as developing models for disease diagnosis or personalized treatment recommendations.

# Potential Future of FL

## Blockchain Integration

We can expect to see FL being integrated with blockchain technology to further enhance the security and privacy of data used in machine learning.

## Smart City

Can be used to analyze data from various sources such as traffic sensors, public transport systems, and energy usage.

## Manufacturing

Can be used in industry to improve quality control and predict equipment failures and trained on data from multiple factories without sharing proprietary information

## Education

Can provide more personalized education, collaborated research

# JUST TO LET YOU KNOW

According to the Health Insurance Portability and Accountability Act, healthcare data breaches in the U.S. have decreased by 48%.

# CONCLUSION

To conclude, I would like to say that federated learning in the field of machine learning has a great potential. I truly believe day by day people will be more aware of their data. Therefore, decentralized machine learning will be applied almost everywhere vastly in healthcare, education, finance and robotics.

# REFERENCE

| | Links / Paper Title |
|---|---|
| 1 | https://en.wikipedia.org/wiki/Federated_learning |
| 2 | https://ai.googleblog.com/2017/04/federated-learning-collaborative.html |
| 3 | https://www.getastra.com/blog/security-audit/healthcare-data-breach-statistics/ |
| 4 | A Performance Evaluation of Federated Learning Algorithms |
| 5 | FEDERATED LEARNING: STRATEGIES FOR IMPROVING COMMUNICATION EFFICIENCY |

# THANK YOU