A Talk for QMUL IADS PhD Forum

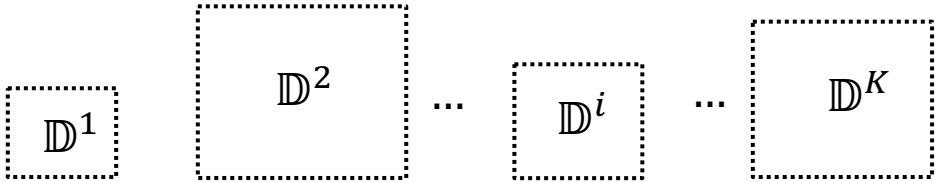
Privacy-Preserving Machine Learning on Distributed Data

Mohammad Malekzadeh
Postdoctoral Researcher
Imperial College London
https://mmalekzadeh.github.io

Where to find data?

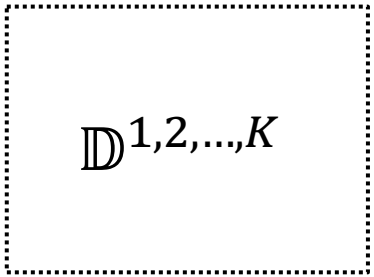
in a silo ⊖ a company/hospital/user i

in distributed silos ✓



in a single location ✗

no-way!



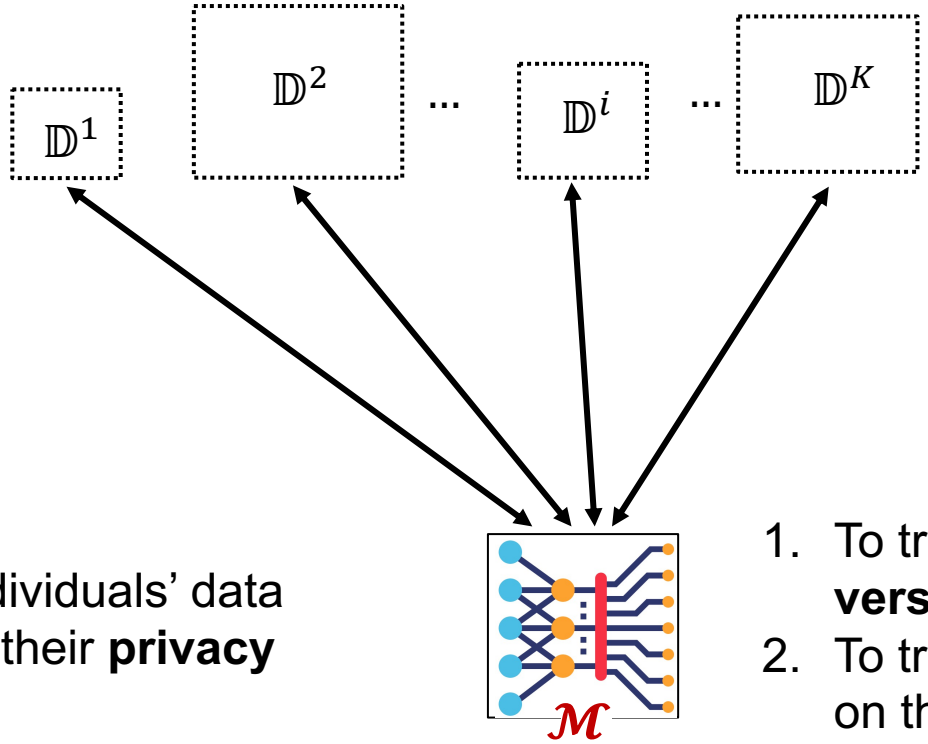
Protecting Privacy in Machine Learning

THREE STAGES:

1. When **collecting** data. (almost) impossible for non-scalar data!
2. When **training** a model on the data.
3. When making **inference** on the data. it is possible!

   Thanks to cryptography, multi-party computations, and/or edge computing.

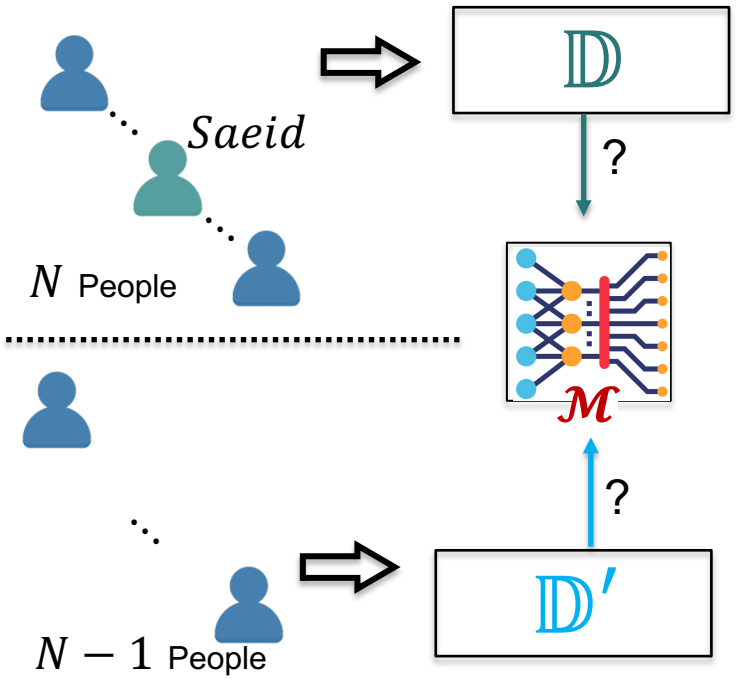
Training



Learning from individuals' data without invading their **privacy**

1. To train on the **sanitized version** of the data.
2. To train a **sanitized model** on the data

Differential Privacy



$$\Pr(\mathcal{M} \text{ is trained on } \mathbb{D}) \leq e^{\epsilon}\Pr(\mathcal{M} \text{ is trained on } \mathbb{D}') + \delta$$

The probability of failure $\ll \frac{1}{|\mathbb{D}|}$

$1 + \epsilon \approx e^{\epsilon}$

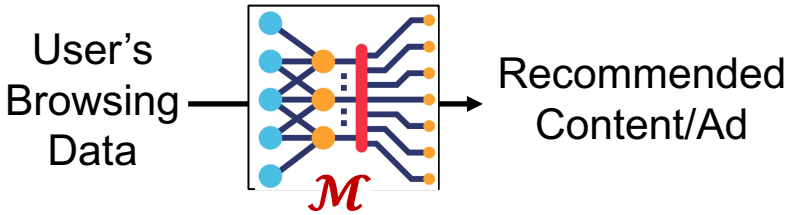
ε=0.1: 1.1 ≈ 1.105 😇
ε=1: 2 ≈ 2.718 😃
ε=2: 3 ≈ 7.389 🙂
ε=3: 4 ≈ 20 🤔
ε=5: 6 ≈ 148 🤨
ε=10: 11 ≈ 22026 😮

By Thomas Steinke

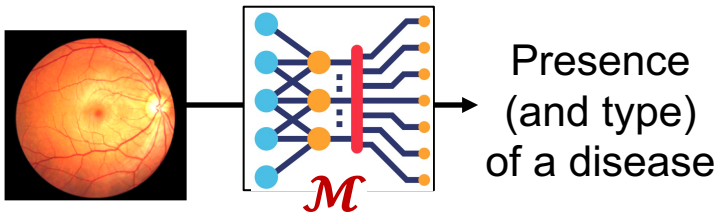
In this talk

How to train **machine learning** models,
while satisfying **differential privacy** for individuals**,**

1. for users of a decentralized system



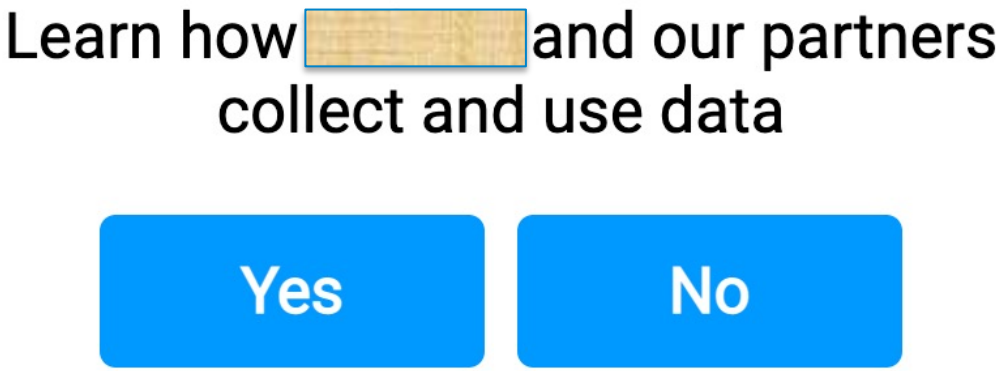
2. on decentralized datasets



1. Privacy-Preserving Personalization

Conference on Machine Learning and Systems (MLSys 2020)

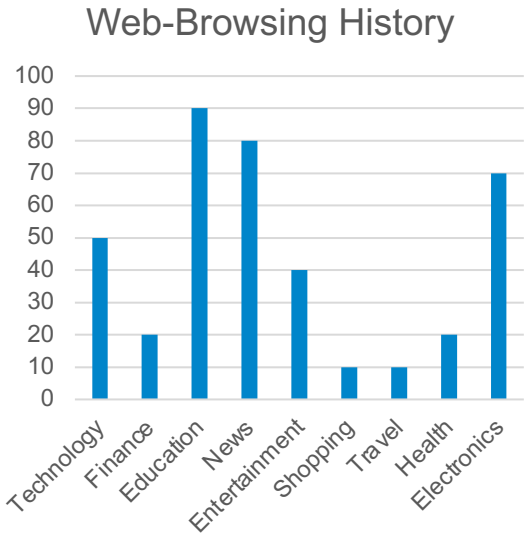
No means No!

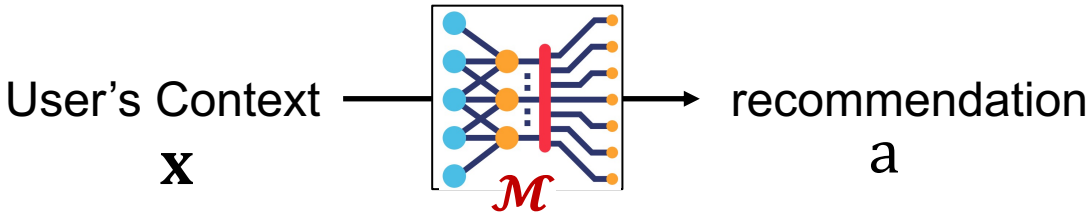

Learn how and our partners collect and use data
Yes
No
Yes, I want to receive personalized ads.
No, I only want to receive non-personalized ads. You can change your choice at any time in our privacy center.

Personalization

"In your world, I have another name. You should know me by it."

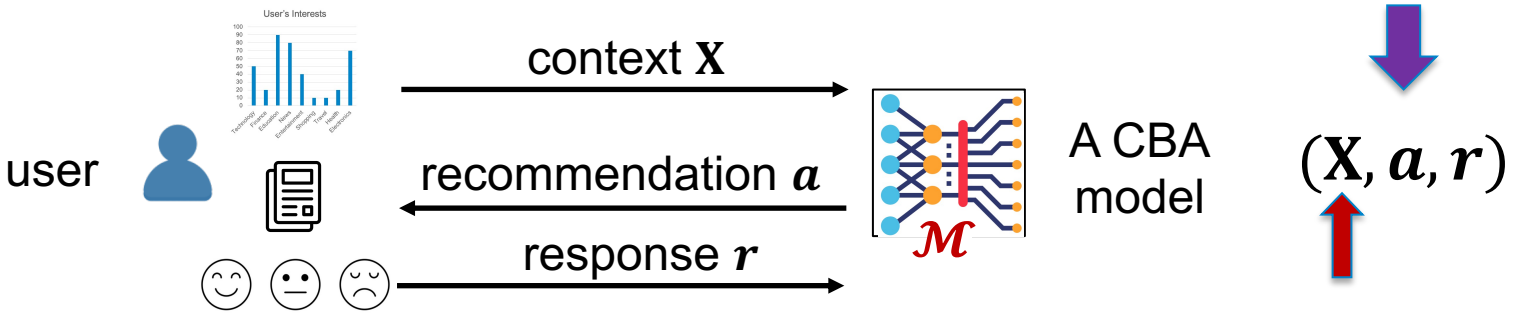
- C.S. Lewis, The Voyage of the Dawn Treader





Algorithm

- Contextual Bandit Algorithms (CBA)
  - to learn a model that maps the user's state to a recommendation that maximizes the reward.



Uniqueness

chrome://site-engagement

Origin	Engagement Score▼
https://www.google.com/	93.01
https://www.overleaf.com/	92.88
https://scholar.google.com/	83.24
https://github.com/	82.12
https://stackoverflow.com/	73.62
https://colab.research.google.com/	69.31
https://soundcloud.com/	65.26
https://ieeexplore.ieee.org/	64.21
https://arxiv.org/	64.03
https://mail.google.com/	56.22
https://medium.com/	46.68
https://tex.stackexchange.com/	33.71
https://stats.stackexchange.com/	33.07
https://www.youtube.com/	30.29
https://keras.io/	30.08
https://en.wikipedia.org/	28.95
https://translate.google.com/	26.76
https://docs.google.com/	26.61
https://towardsdatascience.com/	25.99
https://qmulprod-my.sharepoint.com/	23.66
https://visas-immigration.service.gov.uk/	20.59
https://twitter.com/	18.83
https://www.quora.com/	18.31
https://www.reddit.com/	17.64
https://www.radiojavan.com/	16.55
https://mysis.qmul.ac.uk/	15.91
https://www.gov.uk/	14.93
https://www.tensorflow.org/	14.69
https://drive.google.com/	14.57
https://www.netflix.com/	13.41
https://open.spotify.com/	12.77
https://www.linkedin.com/	12.45

QMUL Students' Website

Google Scholar & Overleaf & Arxiv

TensorFlow & Keras

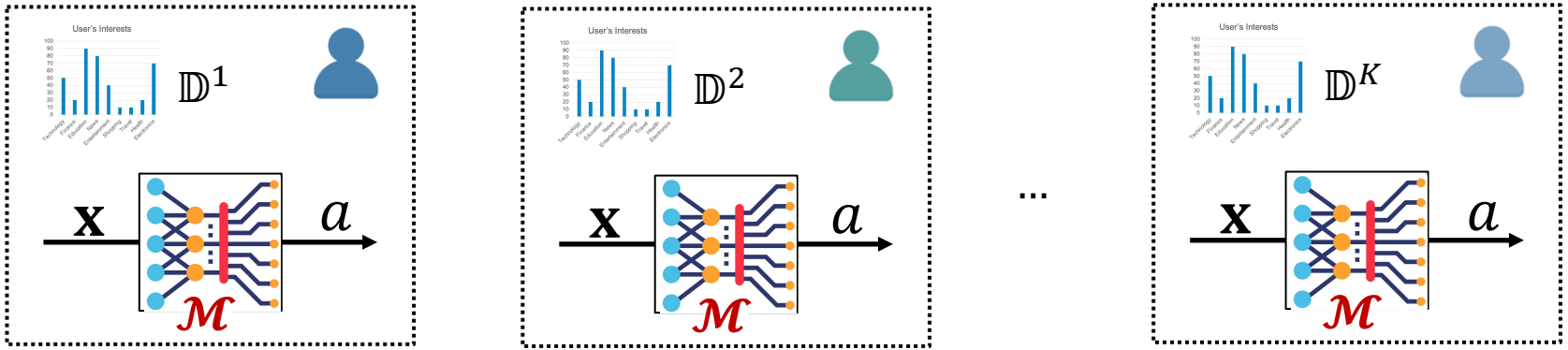
Iranian's Popular Music Streaming

UK Visa

Got it! This is him ☺

On-device Personalization

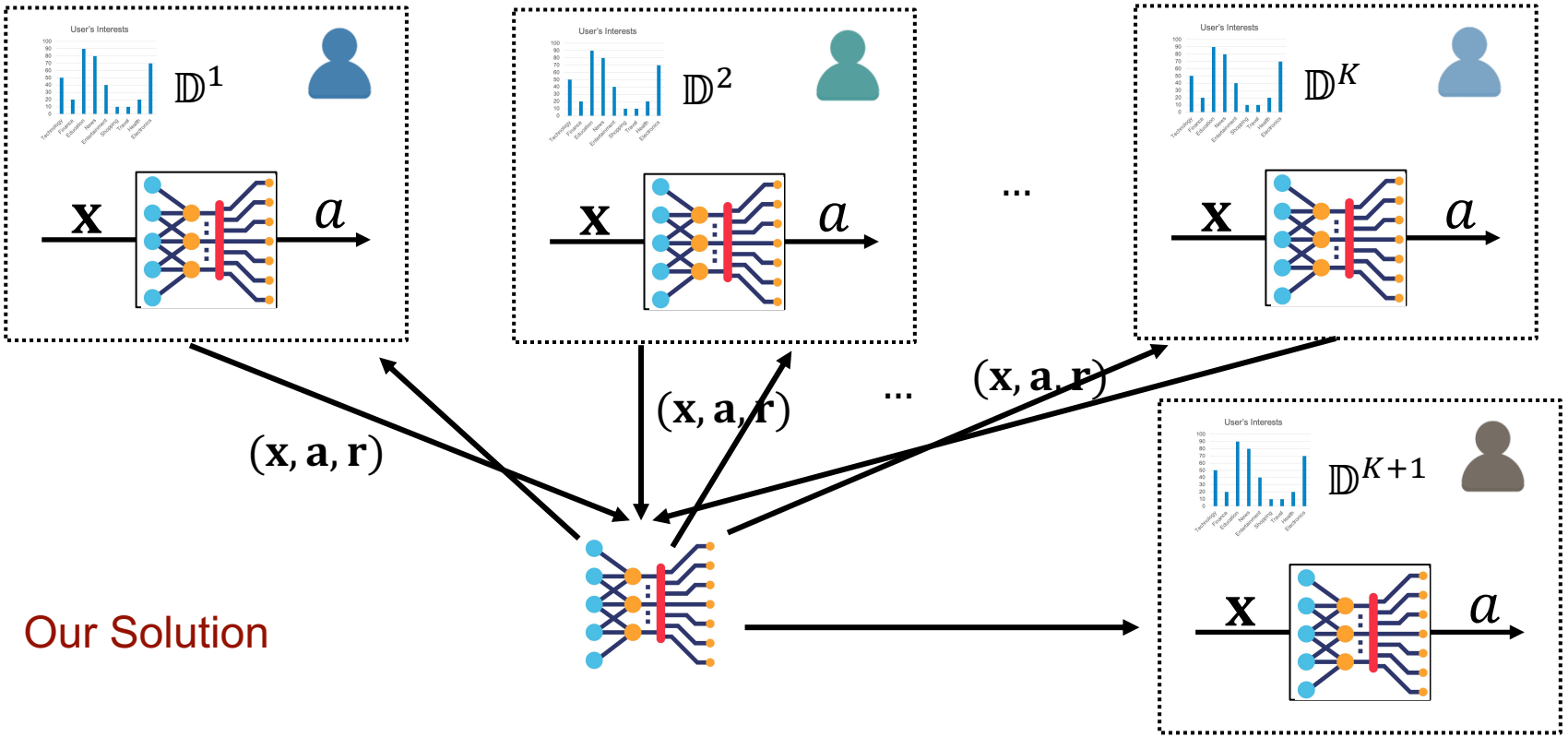
Don’t Be Evil VS Can’t Be Evil



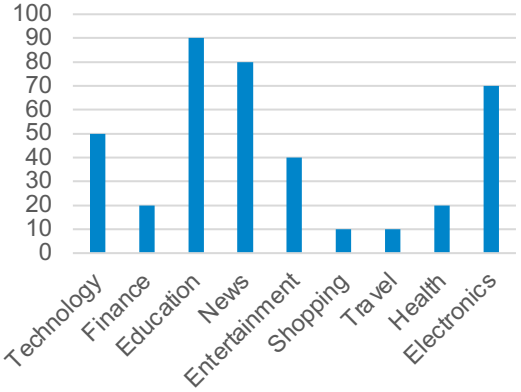
Cold Start Problem

Exploration vs. Exploitation

Collaboration and Warm Start



Step 1: Normalization and Rounding



d-dimensional context vector $\mathbf{x}$

$d = 9$

Normalization (sum=1) and
Rounding (to q decimal digits)

$$N = \binom{10^q + d - 1}{d - 1}$$

stars and bars

.128	.051	.231	.205	.103	.026	.026	.051	.179	$q = 3$
.13	.05	.23	.21	.10	.03	.03	.05	.17	$q = 2$
.1	.1	.2	.2	.1	.0	.0	.1	.2	$q = 1$

$N \approx 25 * 10^{18}$

$N \approx 35 * 10^{10}$

$N = 43758$

The effect of d?

$q = 1$

$$N = \binom{10^q + d - 1}{d - 1}$$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9

$N = 43758$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

$N = 92378$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}

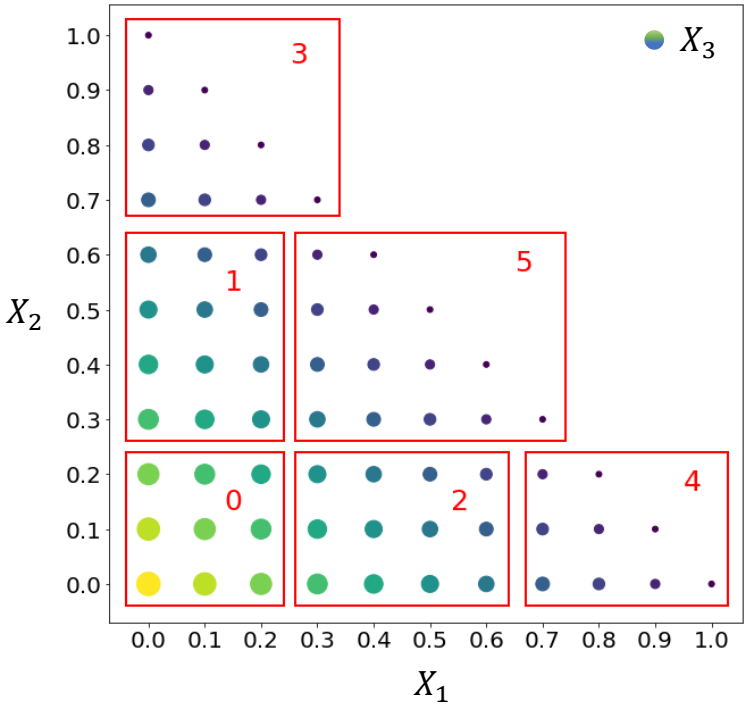
$N = 184756$

Step 2: Clustring

$\mathbf{x}$

x_1	x_2	x_3

d = 3 and q = 1 $N = 66$ possible values for $\mathbf{x}$



Hypothesis: vectors clustered together can be mapped to the same recommendation.

$$\mathbf{x} \rightarrow \boldsymbol{y}$$

$\boldsymbol{k} = 6$ clusters … e.g., by k-means

$$\boldsymbol{N/k} = 11$$

the **average** number of vectors in a cluster.

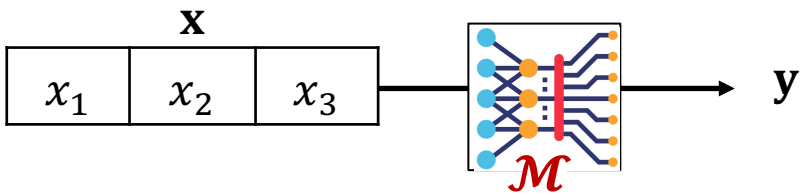
the **minimum** number of vectors in a cluster.

$$\boldsymbol{l} = 9$$

Crowd-Blending Privacy

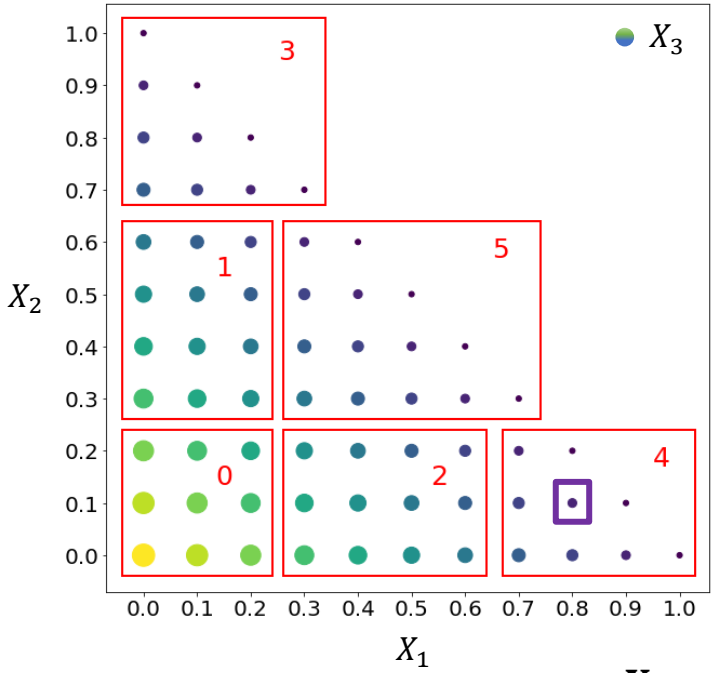
Given $\boldsymbol{l} \geq 1$, we say a data sharing algorithm $\mathcal{M}$ satisfies l-crowd-blending privacy if for all pairs of neighbor datasets $\mathbb{X}$ and $\mathbb{X}'$ differing in only one sample $\boldsymbol{x}$—such that $\mathbb{X} = \mathbb{X}' \cup \{\boldsymbol{x}\}$—we have

$$\texttt{size}\left(\left\{\boldsymbol{y} \in \mathcal{M}(\mathbb{X}) : \boldsymbol{y} = \mathcal{M}\left(\{\boldsymbol{x}\}\right)\right\}\right) \geq \boldsymbol{l} \text{ or } \mathcal{M}(\mathbb{X}) = \mathcal{M}(\mathbb{X}').$$



Gehrke, Johannes, et al. "Crowd-blending privacy." *Annual Cryptology Conference*. Springer, Berlin, Heidelberg, 2012.

Crowd-Blending Privacy



l=9−Crowd−Blending

$\mathbf{y} = \mathbf{4}$

+

Random Sampling

$\mathbf{0 \le p \le 1}$

$\mathbf{x}$

.8	.1	.1

(l-p)–CB to (ϵ,δ)–DP

Differential Privacy

l−Crowd−Blending

y = 4

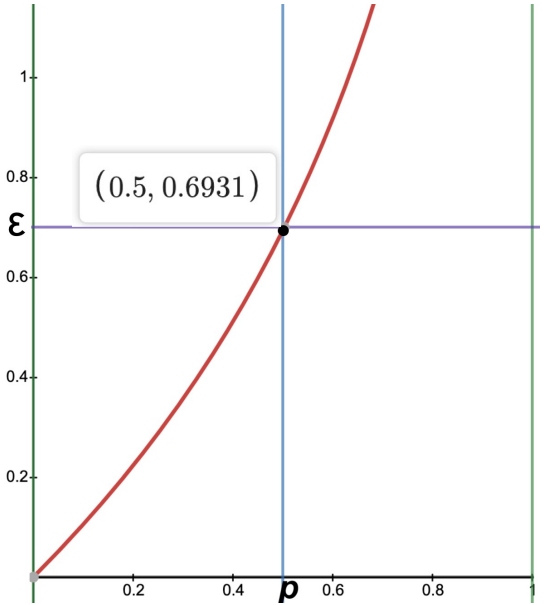
+

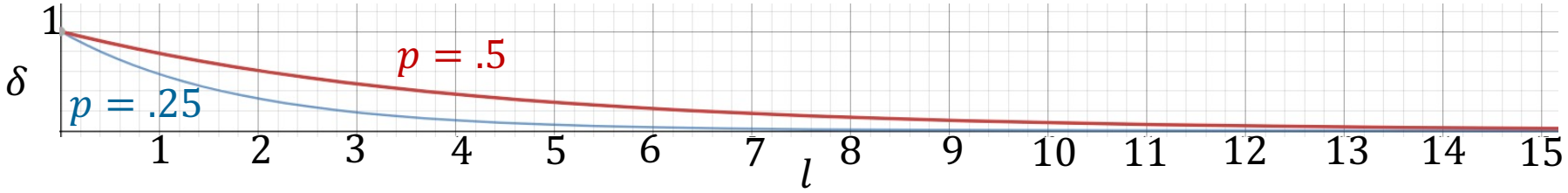
Random Sampling

0 ≤ *p* ≤ 1

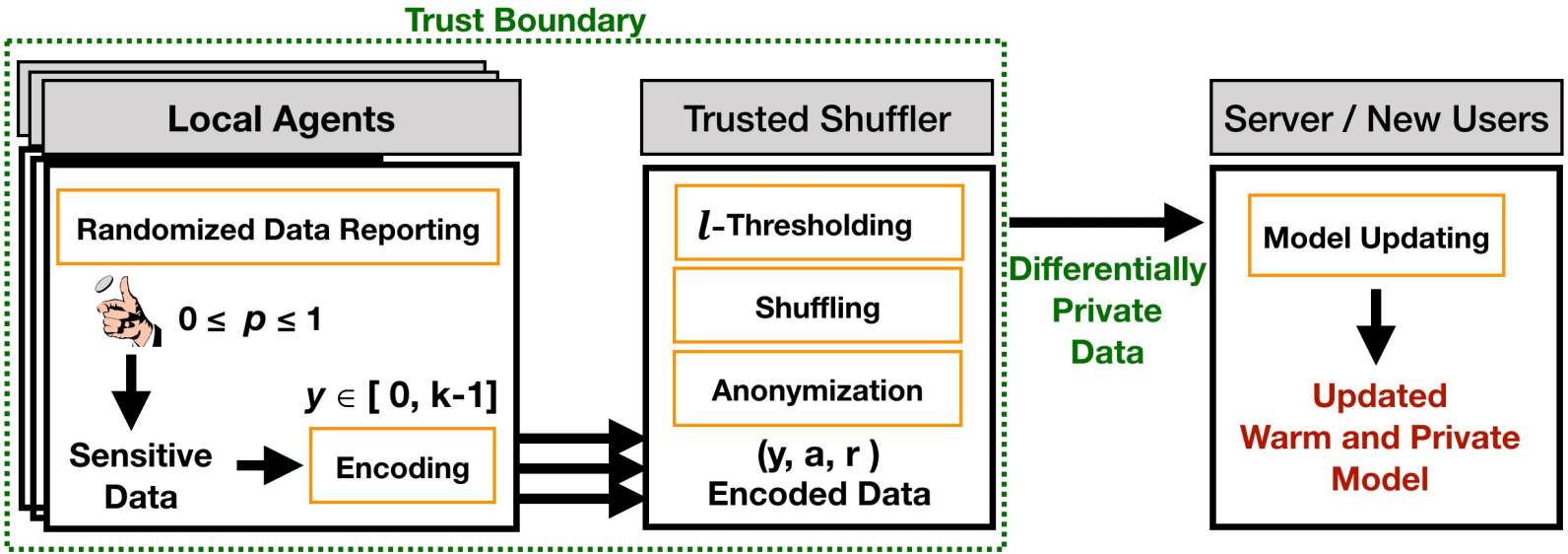
$$\epsilon = \ln\left(p \cdot \left(\frac{2-p}{1-p}\right) + (1-p)\right)$$

$$\delta = e^{-\Omega(l\cdot(1-p)^2)}$$





Big Picture of the Implementation

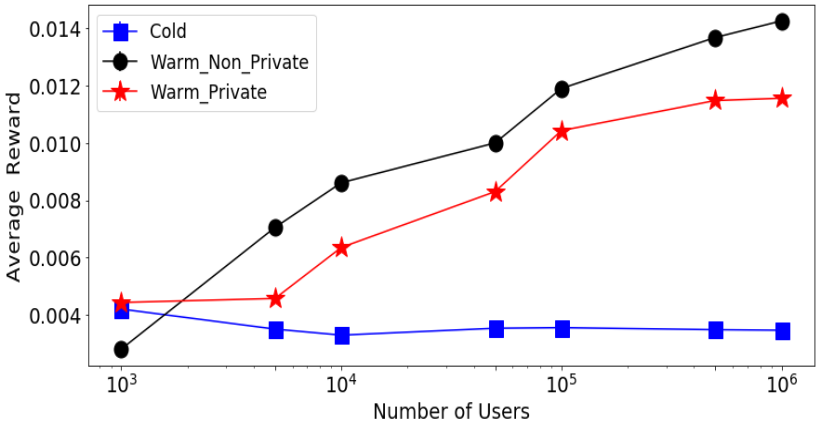


$p = .5$ $l = 10$ $r = \{0,1\}$

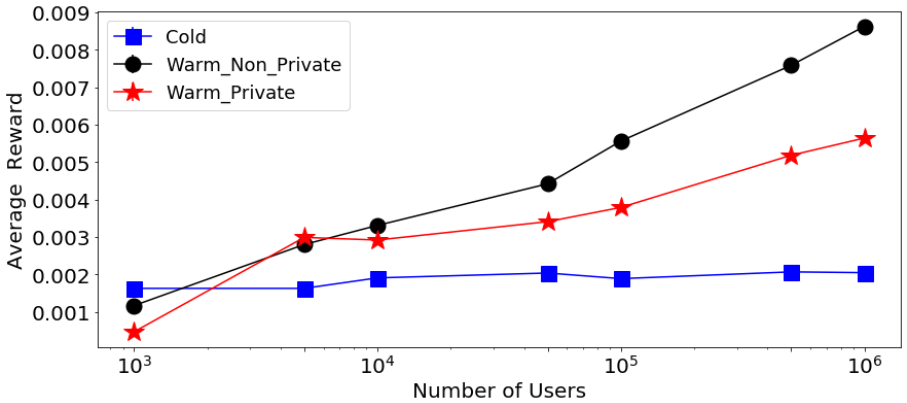
Bittau, Andrea, et al. "Prochlo: Strong privacy for analytics in the crowd." *Proceedings of the 26th Symposium on Operating Systems Principles*. 2017.

Synthetic Recommendation Dataset

$d = 10$ and $k = 1024$



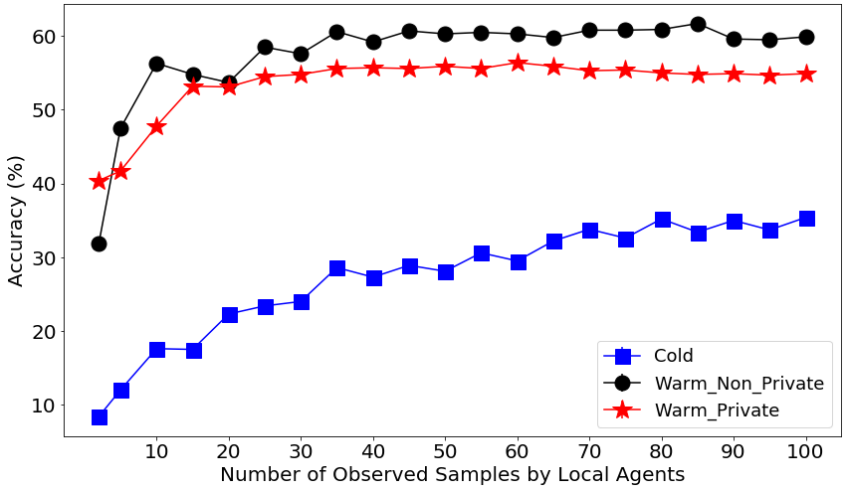
Recommendation categories = **20**



Recommendation categories = **50**

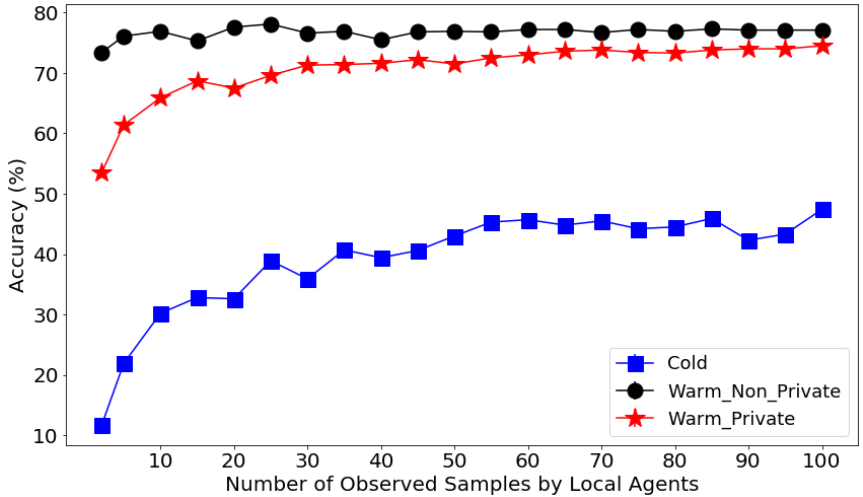
Real-World Multi-Label Dataset

$d = 20$ and $k = 32$



Text-Mining Dataset

labels = **20** users = **200**

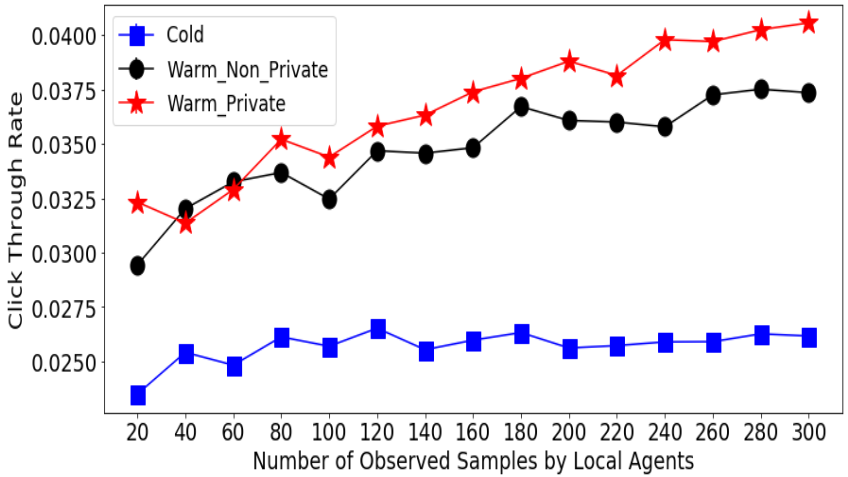


Media-Mill Dataset

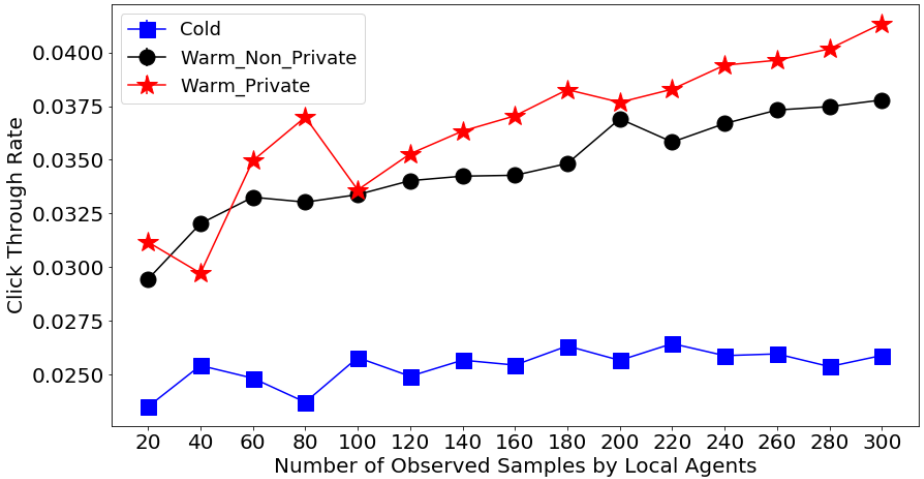
labels= **40** users= **300**

Criteo Online Ad. Dataset

$d = 10$ users= **300** ad. categories = **40**



$k = 32$



$k = 128$

Summary of Part 1

- Private training of a CBA
  - help to achieve warm start and thus better accuracy
  - ensuring differential privacy with an acceptable bound.
  - Negligible computation cost for the edge users. The service provider tolerates the computational cost.

PRIVACY-PRESERVING BANDITS

Mohammad Malekzadeh [1] Dimitrios Athanasakis [2] Hamed Haddadi [2 3] Benjamin Livshits [2 3]

Work done at Brave Research

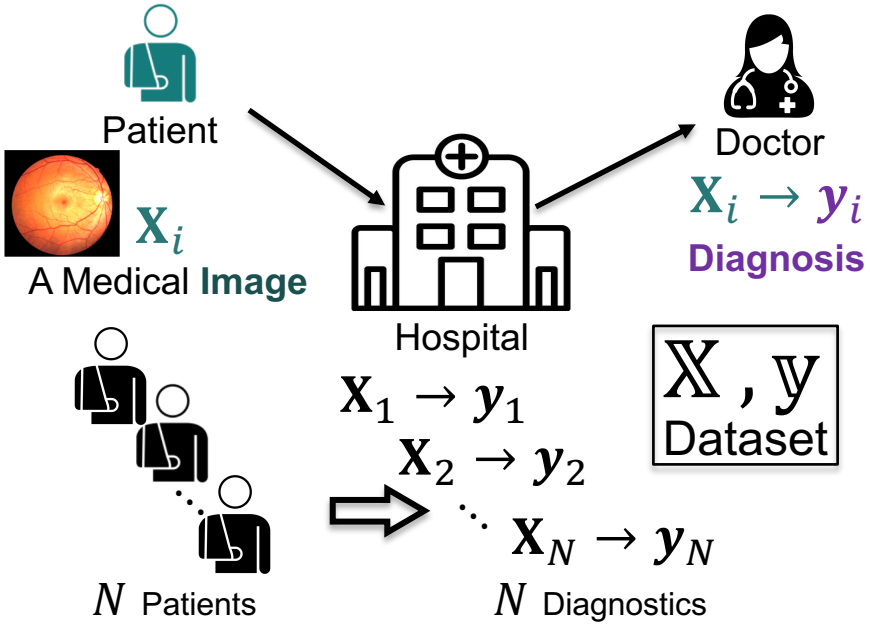
Code:
https://github.com/mmalekzadeh/privacy-preserving-bandits

2. Privacy-Preserving Diagnosis

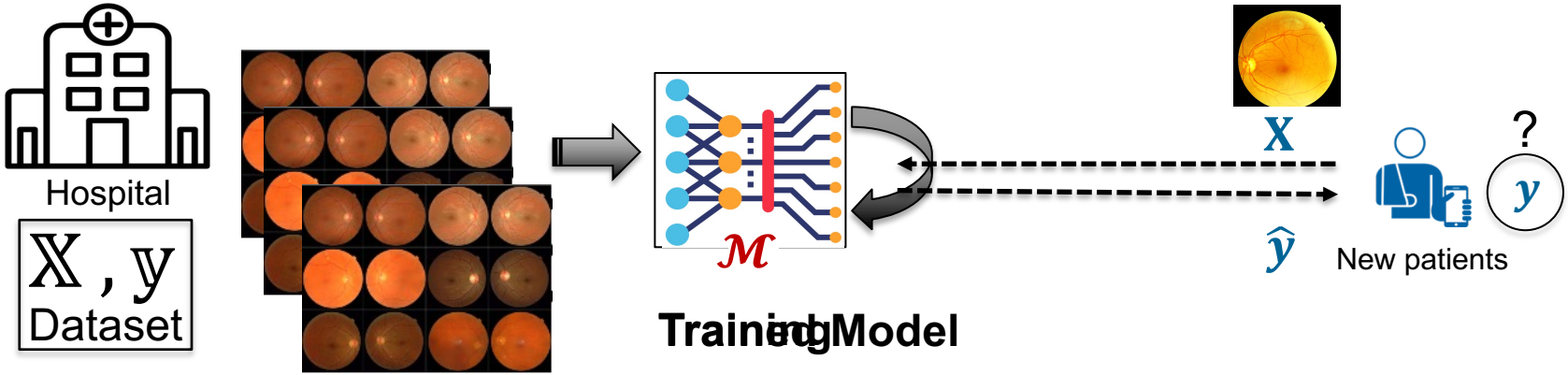
The best submission to the challenge Privacy Preserving AI/ML for Healthcare Applications. ITU's Global Challenge 2020.

AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI-21)

Motivation


Patient
$\mathbf{X}_i$
A Medical Image
Hospital
Doctor
$\mathbf{X}_i \rightarrow \boldsymbol{y}_i$
Diagnosis
$\mathbb{X}$, $\mathbb{y}$
Dataset
$\mathbf{X}_1 \rightarrow \boldsymbol{y}_1$
$\mathbf{X}_2 \rightarrow \boldsymbol{y}_2$
$\mathbf{X}_N \rightarrow \boldsymbol{y}_N$
N Patients
N Diagnostics

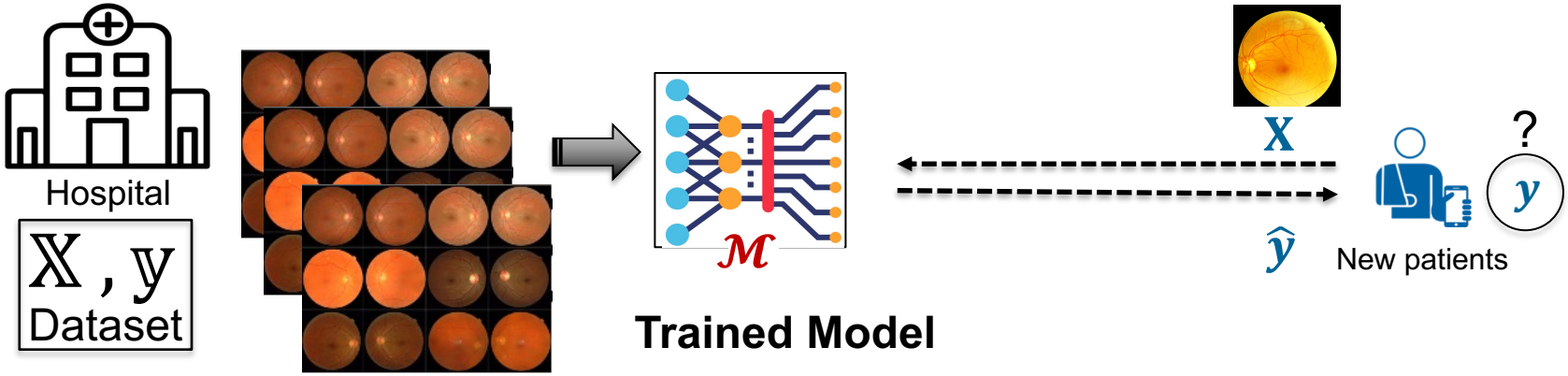
Motivations



Pervasive **Connectivity** enables **Automated Diagnosis**.

- **Coverage**
  - More Patients
  - Rural Area & Developing Countries
- **Efficiency**: Faster and Cheaper diagnosis
- **Lower Burden** on Healthcare System
  - Decision for further examination?
  - Giving Short-Term Advice

Privacy Requirement



Differential Privacy

$\mathcal{M}$ must not reveal the presence (or absence) of any patient i in the training set.

Abadi, Martin, et al. "**Deep learning with differential privacy**." *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016.

DP-SGD

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability $q = L/N$

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

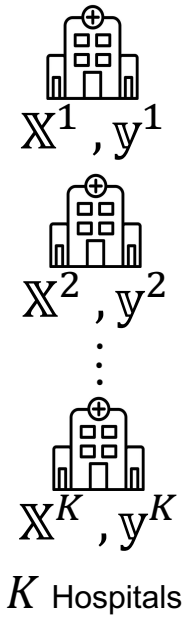
$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} \sum_i (\bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$ with $\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ)

Challenge



Medical Dataset are **Distributed** and Kept **Private**.

Patients' **Privacy** is as important as Patient's **Health**

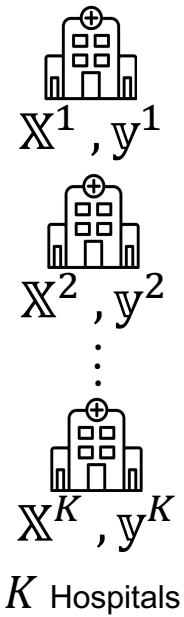
Protected by the **law**!

Federated Learning with DP-SGD

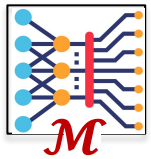
3. Train $\mathcal{M}_1$
With DPSGD

3. Train $\mathcal{M}_2$
With DPSGD

3. Train $\mathcal{M}_K$
With DPSGD

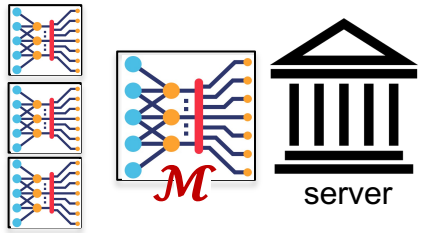


2. Propagate $\mathcal{M}$



4. Aggregation of
$\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$

1. Initialize $\mathcal{M}$



Parallel DP-SGD

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability $q = L/N$

N becomes much smaller, thus larger q!

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} \sum_i (\bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$ with $\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

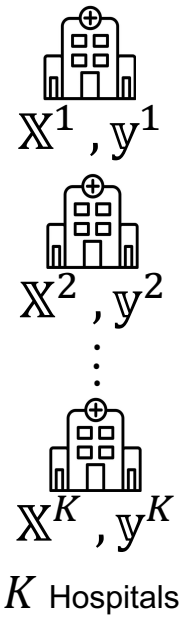
Output θ_T and compute the overall privacy cost (ε, δ)

Parallel DP-SGD with Secure Aggregation

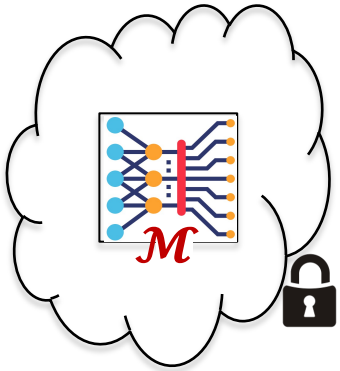
3. Train $\mathcal{M}_1$
With DPSGD
For 1 batch

3. Train $\mathcal{M}_2$
With DPSGD
For 1 batch

3. Train $\mathcal{M}_K$
With DPSGD
For 1 batch

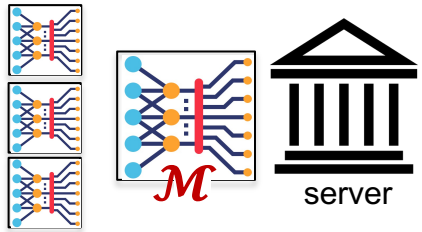


2. Propagate $\mathcal{M}$



4. Secure Aggregation of $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_K$

1. Initialize $\mathcal{M}$



Parallel DP-SGD with Secure Aggregation

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability $q = L/N$

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L} \sum_i \left(\bar{\mathbf{g}}_t(x_i) + \mathcal{N}\left(0, \frac{\sigma^2 C^2 \mathbf{I}}{L}\right)\right)$ with $\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

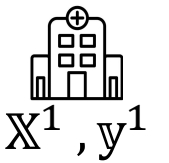
Output θ_T and compute the overall privacy cost (ε, δ)

Implementation(cont.)

3.Train $\mathcal{M}_1$
With DPSGD
For 1 batch

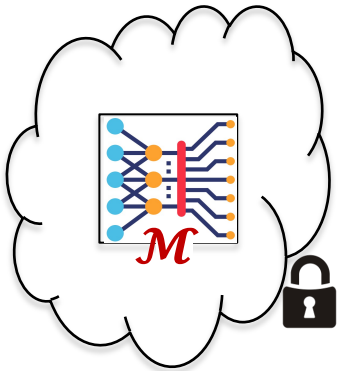
3.Train $\mathcal{M}_2$
With DPSGD
For 1 batch

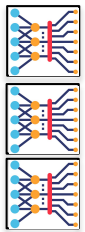
3.Train $\mathcal{M}_K$
With DPSGD
For 1 batch



K Hospitals

2. Propagate $\mathcal{M}$





4.Secure Aggregation of

$\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$

Opacus https://github.com/pytorch/opacus

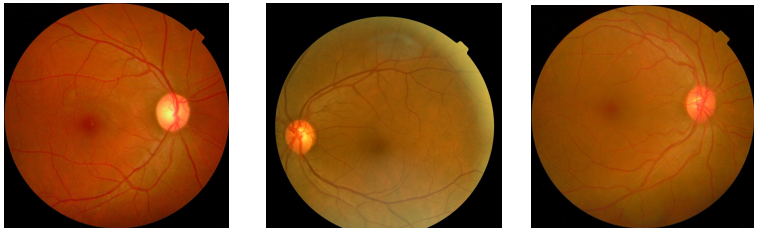
Our Proposed Algorithm

Algorithm 1 *Dopamine*'s Training

Input: K: number of hospitals, $\mathbb{D}$: distributed dataset, $\mathbf{w}$: model's trainable parameters, $\mathcal{L}(\cdot,\cdot)$: loss function, q: sampling probability, σ: noise scale, C: gradient norm bound, η: learning rate, β: momentum, T: number of rounds, (ϵ, δ): bounds on record-level DP loss.
Output: $\mathbf{w}_G$: optimized global model.
$\mathbf{w}_G^0 =$ random initialization.
$\hat{\epsilon} = 0$
for $t : 1, \ldots, T$ **do**
  for $k : 1, \ldots, K$ **do**
    $\mathbb{D}_k^t = Sampling(\mathbb{D}_k)$ // by uniformly sampling each item in $\mathbb{D}_k$ independently with probability q.
    for $\mathbf{x}_i \in \mathbb{D}_k^t$ **do**
      $\mathbf{g}^t(\mathbf{x}_i) = \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_G^{t-1}, \mathbf{x}_i)$
      $\bar{\mathbf{g}}^t(\mathbf{x}_i) = \mathbf{g}^t(\mathbf{x}_i) / \max\left(1, \frac{||\mathbf{g}^t(\mathbf{x}_i)||_2}{C}\right)$
    end for
    $\tilde{\mathbf{g}}_k^t = \frac{1}{|\mathbb{D}_k^t|}\left(\sum_{\mathbf{x}_i \in \mathbb{D}_k^t} \bar{\mathbf{g}}^t(\mathbf{x}_i) + \mathcal{N}(0, \frac{\sigma^2 \cdot C^2 \cdot \mathbf{I}}{K})\right)$
    $\hat{\mathbf{g}}_k^t = \tilde{\mathbf{g}}_k^t + \beta \hat{\mathbf{g}}_k^{t-1}$ // $\hat{\mathbf{g}}_k^0 = 0$
    $\mathbf{w}_k^t = \mathbf{w}_G^{t-1} - \eta \hat{\mathbf{g}}_k^t$
  end for
  $\hat{\epsilon} = CalculatePrivacyLoss(\delta, q, \sigma, t)$ // by Moments Accountant (Abadi et al. 2016)
  if $\hat{\epsilon} > \epsilon$ **then**
    return $\mathbf{w}_G^{t-1}$
  end if
  $\mathbf{w}_G^t = \frac{1}{K}\left(SecureAggregation(\sum_k \mathbf{w}_k^t)\right)$
  $Broadcast(\mathbf{w}_G^t)$
end for

Evaluation

- **Dataset:**
  - **Diabetic Retinopathy**[4]
  - **Five** Classes: *normal*, *mild*, *moderate*, *severe*, and *proliferative*.
  - **3662** images: 2931 for training, 731 for testing.
  - Dimensions: **224×224**



- **Deep Neural Network:**
  - **SqueezeNet**[5]
  - 50x **fewer** parameters than the famous AlexNet.
  - Yet, achieves the **same** level of AlexNet's accuracy on ImageNet.

- **Simulation:**
  - **10** hospitals and **1** server
  - data distributed **i.i.d** and **equal**
  - **100** epochs, batch size **50**
  - $\delta - DP = \mathbf{0.0001}$

[4] Choi, J. Y.; et. al. . 2017. Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database. PLOS ONE12: 1–16.

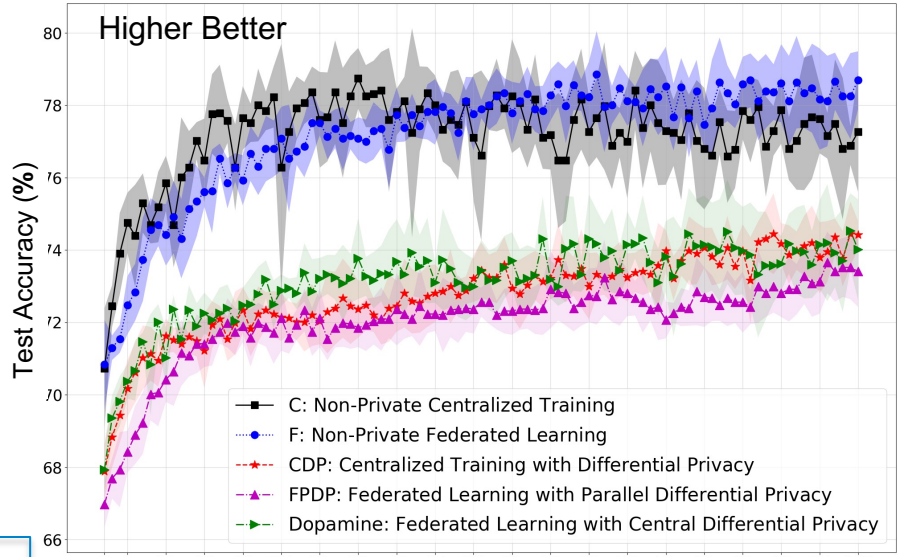
[5] Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." *arXiv preprint arXiv:1602.07360* (2016).

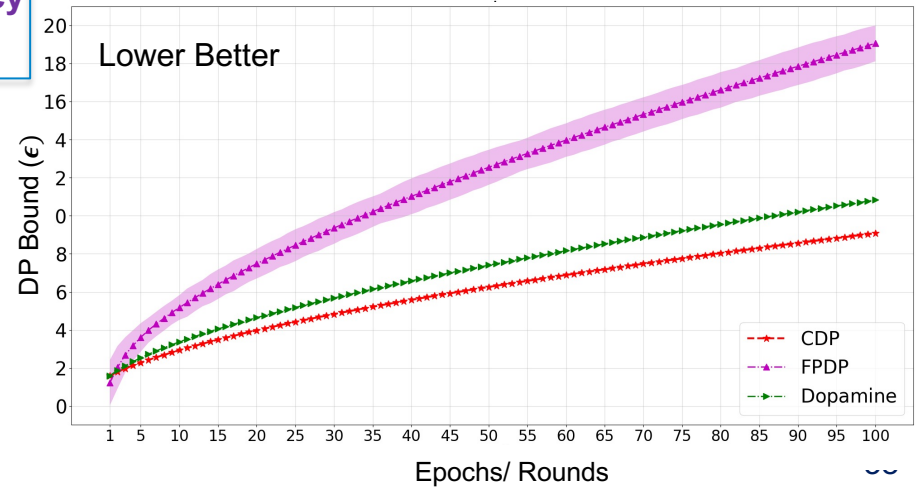
Experimental Results

Baselines:

1) **Centralized Learning without Privacy**
2) **Federated Learning without Privacy**
3) **Centralized Learning with Differential Privacy**
4) **Federated Learning with Parallel Differential Privacy**
5) **Our Solution**

- **(1)** & **(3)** are not achievable in practice (e.g. due to law)
- **(2)** & **(4)** does not provide any (meaningful) privacy
- **Our Solution** achieves the best **utility-privacy trade-off.**





Contributions

1. **Federated learning** on **DNNs** with **patient-level DP** on a **medical dataset** (open-sourced: https://github.Com/ipc-lab/private-ml-for-health)
2. **Momentums** in **federated DP-SGD** achieving **better accuracy & stable training**
3. Achieved the best **utility-privacy trade-off**, among other alternatives.

In Progress

1. End-to-end Secure Aggregation Using Homomorphic encryption
2. Further Evaluation: Other datasets --- Other DNNs.
3. Keeping the trained Model Private at the Server's Side.

Open Research Questions

1. More accurate and efficient FL algorithms with DP.
2. When patients could have more than one sample data.

Summary of Part 2

- Federated Learning on DNNs with Patient-Level DP on a Medical Dataset
- First to use Momentums in Federated DP-SGD for achieving Better Accuracy & Stable Training
- Achieved the best Utility-Privacy Trade-off, among other alternatives.

Dopamine: Differentially Private Federated Learning on Medical Data

Mohammad Malekzadeh, Burak Hasircioglu, Nitish Mital, Kunal Katarya, Mehmet Emre Ozfatura, Deniz Gündüz*

Department of Electrical and Electronic Engineering, Imperial College London.
{m.malekzadeh, b.hasircioglu18, n.mital, kunal.katarya15, m.ozfatura, d.gunduz}@imperial.ac.uk

https://arxiv.org/abs/2101.11693

Code:

https://github.com/ipc-lab/private-ml-for-health