Privacy Evaluation and Accuracy for Different ML Secure Models

Privacy Project

Anastasiya Merkushova
Alessandro Stanghellini
Yannick Martin

University Basel

April 2024

Introduction

- ML in medicine
- Sensitive data
- Privacy Preserving Training techniques
- Membership Inference Attacks

Dataset

- Texas100
- Technical Dataset
- 67330 records
- 6169 binary features (information about the patient, the causes of injury, the diagnosis)
- 101 classes which represent the most frequent medical procedures

DP-Libraries for Machine Learning

PyVacy

- Privacy Algorithms for PyTorch (DPSGD)
- Not well maintained

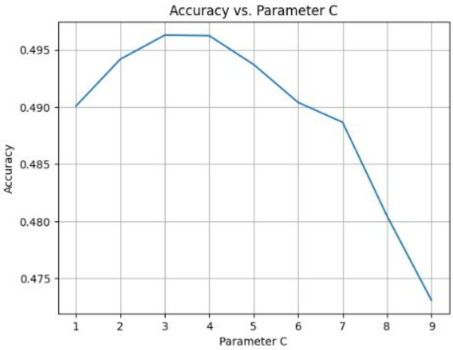
Opacus

- Privacy Algorithms for PyTorch (DPSGD)
- User-Friendly

Algorithm 3 Setting up Opacus

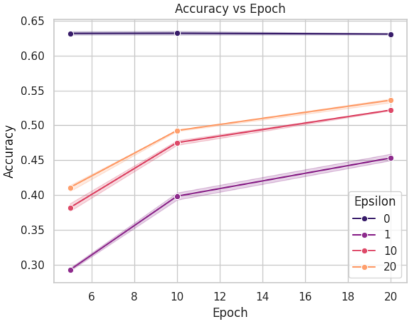
```
define your components(network, optimizer) as usual
Initialize Privacy Engine:
privacy_engine = PrivacyEngine()

Make Network Private with Epsilon:
network, optimizer, trainloader =
  privacy_engine.make_private_with_epsilon(
    module=network,
    optimizer=optimizer,
    data_loader=trainloader,
    max_grad_norm=C,
    target_epsilon=epsilon,
    target_delta=Delta,
    epochs=epochs
)
Now Start the training as usual
```

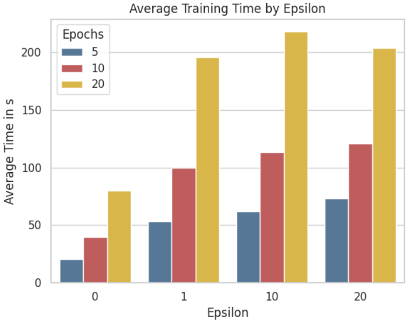
Hyper parameter finding & Results



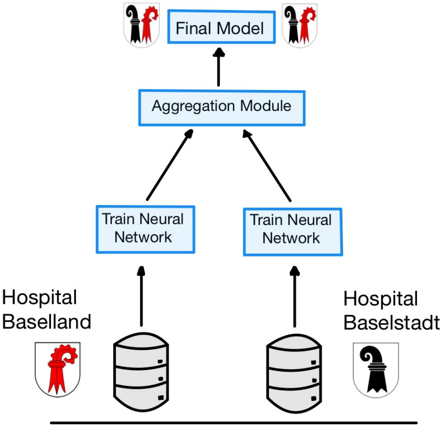
Model	Accuracy	Epsilon	Epochs
NN	0.63	-	10
Opacus	0.55	20	20
Opacus	0.53	10	20
Opacus	0.45	1	20
PyVacy	0.37	28	20
PyVacy	0.38	12	20
PyVacy	0.26	1.3	20

Results

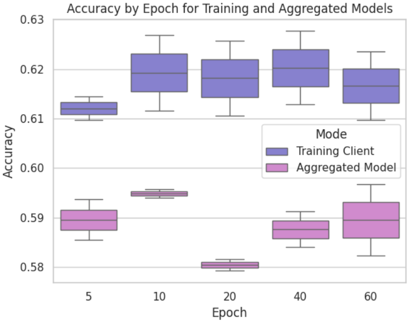

Accuracy vs Epoch
Accuracy
Epoch
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
6
8
10
12
14
16
18
20
Epsilon
0
1
10
20

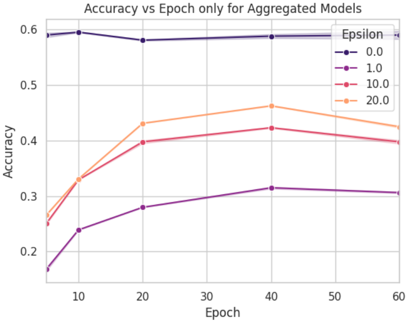

Average Training Time by Epsilon
Average Time in s
Epsilon
Epochs
5
10
20
200
150
100
50
0
0
1
10
20

Federated Learning

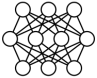


Results from Federated Learning





Membership Inference

Was  **trained**

on the example  **?**

Why?

Curiosity!

Reconnaissance!

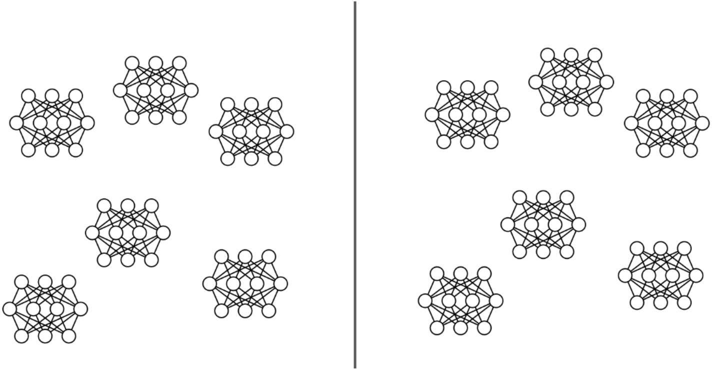
Data Extraction!

Auditing!

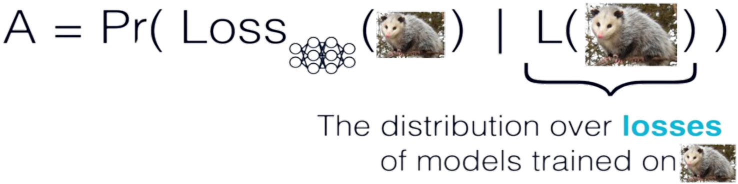
Attack via Population Overview

Models trained on :

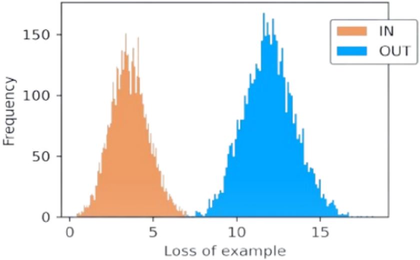
Models not trained on :



Membership Inference



Membership Inference



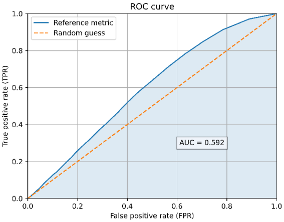
Privacy Meter: Attack via Population Data

- Privacy Meter [1] library for auditing data privacy in ML algorithms.
- Population attack uses direct statistical analysis of the target model, avoiding shadow models.
- Attack thresholds are tailored to each target model, ensuring robustness across varied datasets.
- Empirically, attackers approximate distribution by sampling records from the population data pool.

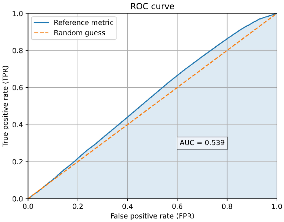
Privacy Meter: Attack via Population Data

Metric Results Generate report	Metric Results: save the attack performance Save TPR, FPR, ROC, AUC, etc.
Metric Different attacks	Metric: launch membership inference attack Given the auditing dataset and auditing reference models, find the threshold for signals to determine the member for the target dataset
Information Source (Dataset, Models, Signals)	Information source: define the attack setting 1. Target model 2. Target dataset 3. Auditing dataset 4. Auditing reference models 5. Membership inference attack signal (e.g., loss)

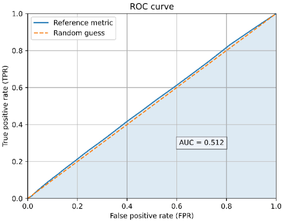
Privacy Meter: Attack via Population Data



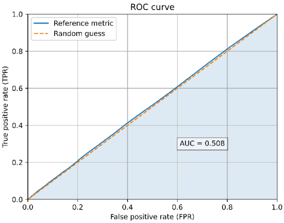
(a)



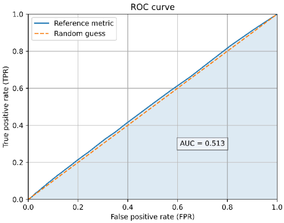
(b)



(c)



(d)



(e)

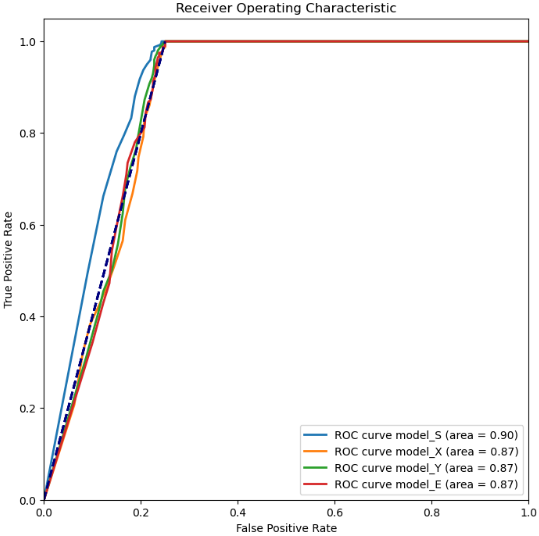
Membership Inference Attack

- Why can they be effective?
- Black-box and white-box attack

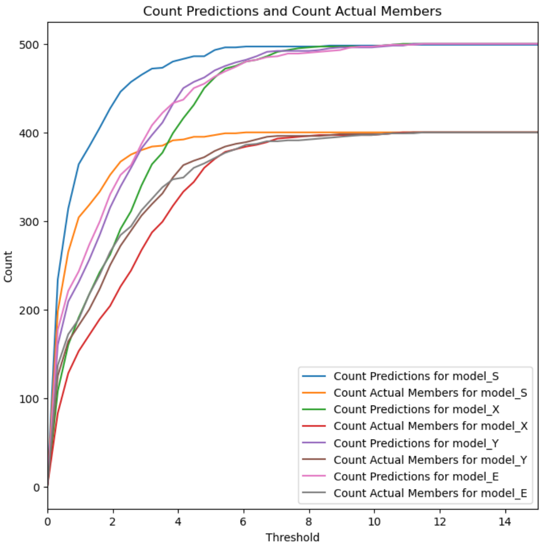
Attacks

- Black-box attack using loss
- White-box attack using norm of the gradient of the loss function with respect to input point

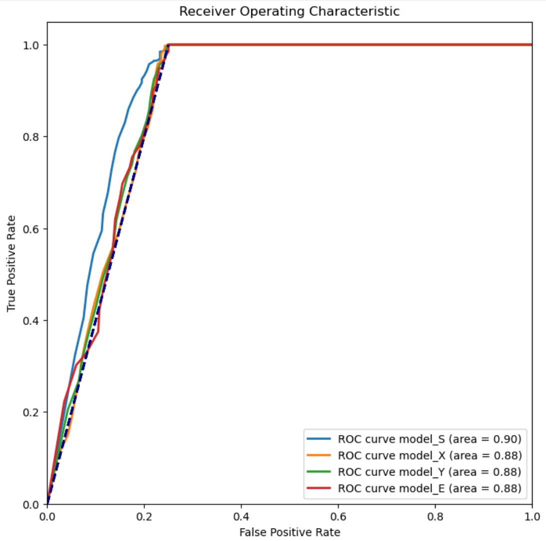
Black-box attack



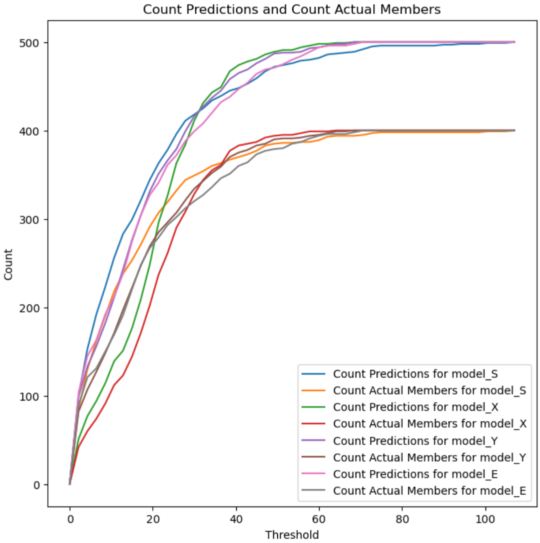
Black-box attack cont'd



White-box attack



White-box attack cont'd



Take-home messages

- We trained a model in private and non-private way
- Privacy-Meter investigation

Thank you for listening

References

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri.
Enhanced membership inference attacks against machine learning models, 2022.

Privacy Meter: Attack via Population Dataset

- The hypothesis test with model-dependent attack threshold:

$$\text{If } \ell(\theta, x_z, y_z) \leq c_\alpha(\theta), \text{ reject } H_0$$

- Out world:

$$P_{out}(D, \theta, z) : D = D_0, \theta = \theta_0, z \sim \pi$$

- Empirical distribution approximation:

$$p_{\theta_0} = \{(\theta_0, z_i)\}_{i=1,2,\ldots}, \text{ and } z_1, z_2, \ldots \sim \pi$$

- Attack threshold for low false positive rate:

$$\frac{\left((\theta, z) \in p^{\theta)0} : \ell(\theta, x_z, y_z) \leq c_\alpha(\theta_0)\right)}{|p^{\theta_0}|} = \alpha$$