Empirical Privacy Risk Analysis on Differentially-private Machine Learning

Contents

1.Introduction

- **Problem Statement:**
  - In normal training, information of training data will be encoded in the gradients and further into the model parameters.
  - Differential privacy (DP) [1] is a mathematically rigorous definition and differentially private machine learning (DPML) provides a *theoretical* lower-bound guarantee for data privacy.
  - However, it is unknown if DPML has desired protection of data privacy *in practice*.

- **Contribution:**
  - Empirically measured the actual effects of DP in protecting privacy
  - Used Membership Inference Attack (MIA) [2] to measure level of privacy protection
  - Experimented on state-of-the-art MIA and introduced a novel attack

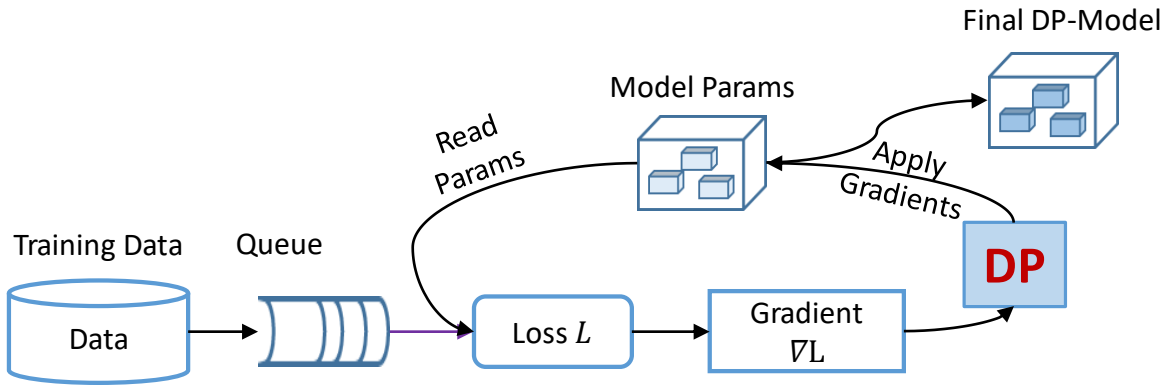
Contents

2.1 Differentially Private Machine Learning

- **General Approach of DPML:**

  Add noise in the gradient descent step during training. A specific level of the added noise yields a corresponding (ϵ, δ)-differentially private neural network.

- **Workflow of Gradient perturbation based DP training**



Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

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

Take a random sample L_t with sampling probability 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

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

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

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

[6]

- **Some variants of DPML:**
  - constant gradient clipping and noise
  - noise decay
  - adaptive clipping bound

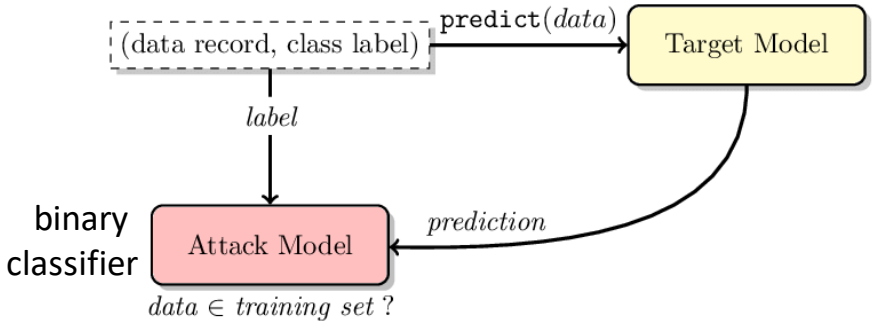
Compare with standard gradient descent:

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \nabla_{\theta_t} L(\theta_t, x)$$

2.2 Membership Inference Attack

- **Membership Inference Attack (MIA):**

  The goal of the attack is to determine whether a given data record was part of the model's training set or not, which is an indication of information leakage through the model. The attacker can use the output of the trained models (e.g. loss, logits, prediction vectors) to perform the attack.

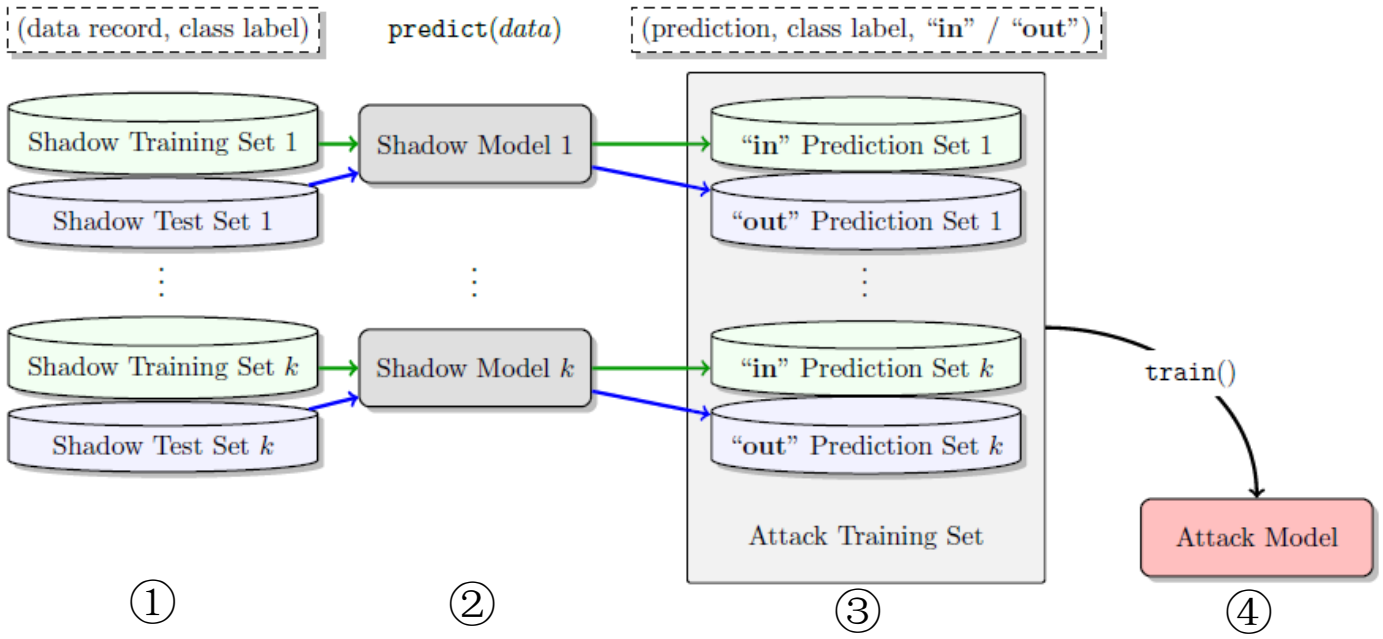


E.g:
knowing that a certain patient's clinical record was used to train a model associated with a disease (e.g, to determine the appropriate medicine dosage or to discover the genetic basis of the disease) can reveal that the patient has this disease.

- **Two settings of MIA:**
  - Black-box - Shadow model MIA [2]: train shadow models which simulate the target model, and the attack model is trained on the outputs of shadow models
  - White-box - Tensorflow MIA [3]: attack model trained directly on the outputs of the target model on its training/test set

2.3 Shadow Model MIA

- **Knowledge of the attacker:**
  - Black box access to the target model
  - Architecture of the target model (no knowledge of model weights)
  - No access to the real training set (only the distribution)
- **Pipeline of the approach**

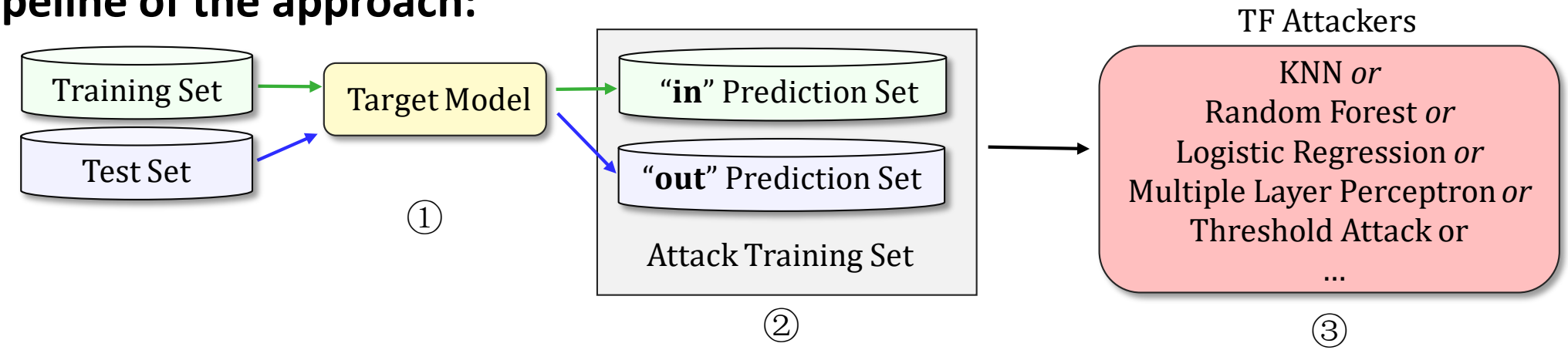


Steps of Shadow MIA:

1. Develop shadow dataset which resembles the actual dataset and split the shadow dataset into shadow training/test set
2. Train shadow models (same architecture as the target model) on shadow training set
3. Construct attack training set:
   For data x_1 "in" the shadow training set, the attack train data is $(ShadowModel(x_1), \mathbf{1})$;
   For data x_2 "out" of shadow training set: $(ShadowModel(x_2), \mathbf{0})$
4. Train the attack model on the attack training set

2.4 Tensorflow MIA

- **Knowledge of the attacker:**
  - No knowledge of the target model architecture/weights
  - Output of training data and test data on the target model
- **Pipeline of the approach:**

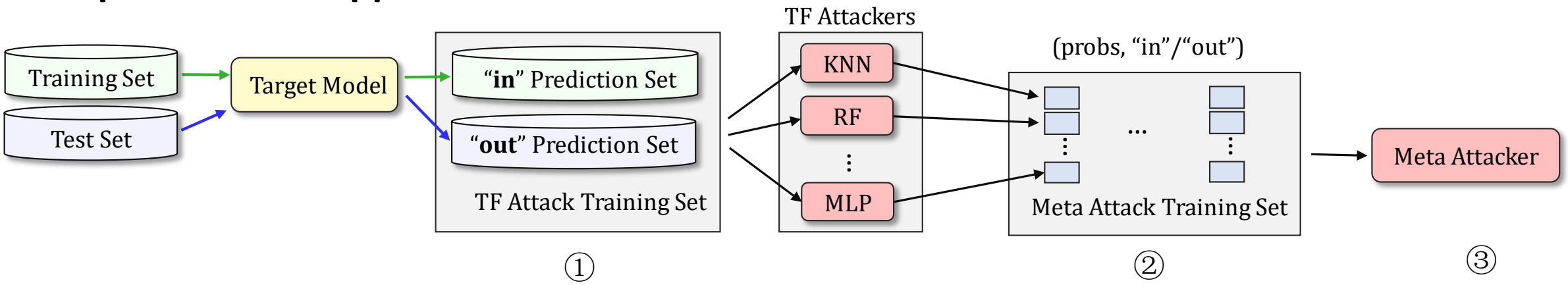


Steps of Tensorflow MIA:

1. Train target model on training set
2. Construct attack training set:
   For data x_1 "in" the training set, the attack train data is (Target$Model(x_1)$, **1**);
   For data x_2 "out" of the training set: (Target$Model(x_2)$, **0**)
3. Train the attack model (e.g. KNN, TF, MLP …) on the attack training set

2.5 Meta Attack

- **Meta Attack on TF attackers**:
  Use a neural network as an higher-level attacker, which takes in the output of the TF attackers (KNN, RF, LR, MLP…) and outputs the membership prediction.
- **Pipeline of the approach:**



Steps of Meta Attack:

1. Train TF Attackers (KNN, TF, MLP) as in TF MIA
2. Construct meta attack training set by concatenating probability outputs of TF Attackers:
   For data x_1 "in" the training set, the meta attack train data is $([KNN(x_1), RF(x_1), \dots, MLP(x_1)]$, **1**)
   For data x_2 "out" of the training set: $([KNN(x_2), RF(x_2), \dots, MLP(x_2)]$, **0**)
3. Train the meta attacker(neural network) on the meta attack training set

Contents

3.1 Experiment Settings

- **Target Model**:

  LeNet

- **Dataset**:

  CIFAR10, CIFAR100, MNIST

- **Metrics for MIA**:

  Accuracy, precision, F1 score, advantage*

***Advantage[5]**:

$$adv = P(A = 0|b = 0) + P(A = 1|b = 1) - 1$$

where b: if a data record belongs to the trainset

A: the guess of the adversary about if the data belongs to trainset

A random adversary would have advantage of 0, and a perfect adversary 1.

- **DPML configuration:**
  - Train: test = 1:1 in training target model
  - Split train/test randomly for 3 times and train with each split to get averaged results
  - Choose noise multiplier $\sigma \in \{0.5, 1.0, 5.0\}$ to achieve different ϵ
  - Choose $\delta = 10^{-6}$, clipping bound $C = 1.0$, use constant gradient clipping and noise
  - Keep all the other settings (learning rate, epoch, batch size, etc) in DP training same as in normal training

3.1 Experiment Settings

- **Shadow Model MIA:**
  - 4 shadow models
  - Attacker is a fully connected neural network
  - Train an attacker for each data class
- **Tensorflow MIA:**
  - Attackers include Random Forest, Logistic Regression, MLP, SVM, Gaussian Process Classifier, Decision Tree, AdaBoost, threshold attack
  - For each type of attacker, exhaustively search with a 3-fold cross validation over specified hyper-parameters and pick out the best hyper-parameter
  - Train an attacker for each data class
- **Meta Attack MIA:**
  - Same settings as in Tensorflow MIA
  - Meta attacker is a fully connected neural network
  - Train an attacker for each data class

3.2 DPML Results

Tab. Train and test accuracy for models trained under different settings

dataset		No dp	noise=0.5	noise=1.0	noise=5.0
MNIST	ϵ	-	45.6	5.7	0.7
	train	0.991	0.909	0.880	0.775
	test	0.989	0.939	0.926	0.850
CIFAR10	ϵ	-	49.7	6.2	0.8
	train	0.815	0.587	0.531	0.410
	test	0.738	0.594	0.542	0.425
CIFAR100	ϵ	-	49.7	6.2	0.8
	train	0.513	0.239	0.183	0.0937
	test	0.42	0.245	0.187	0.0943

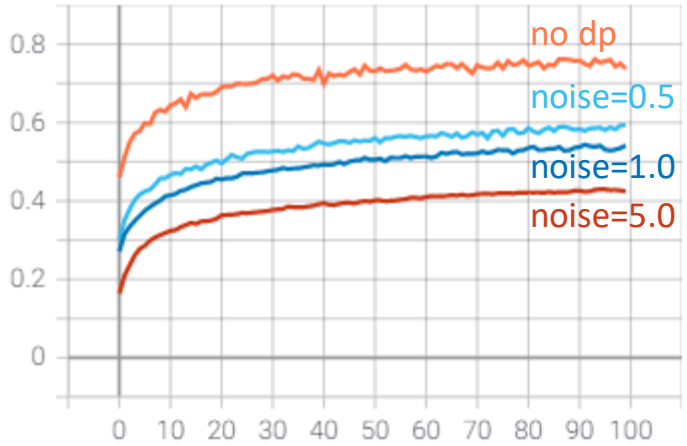
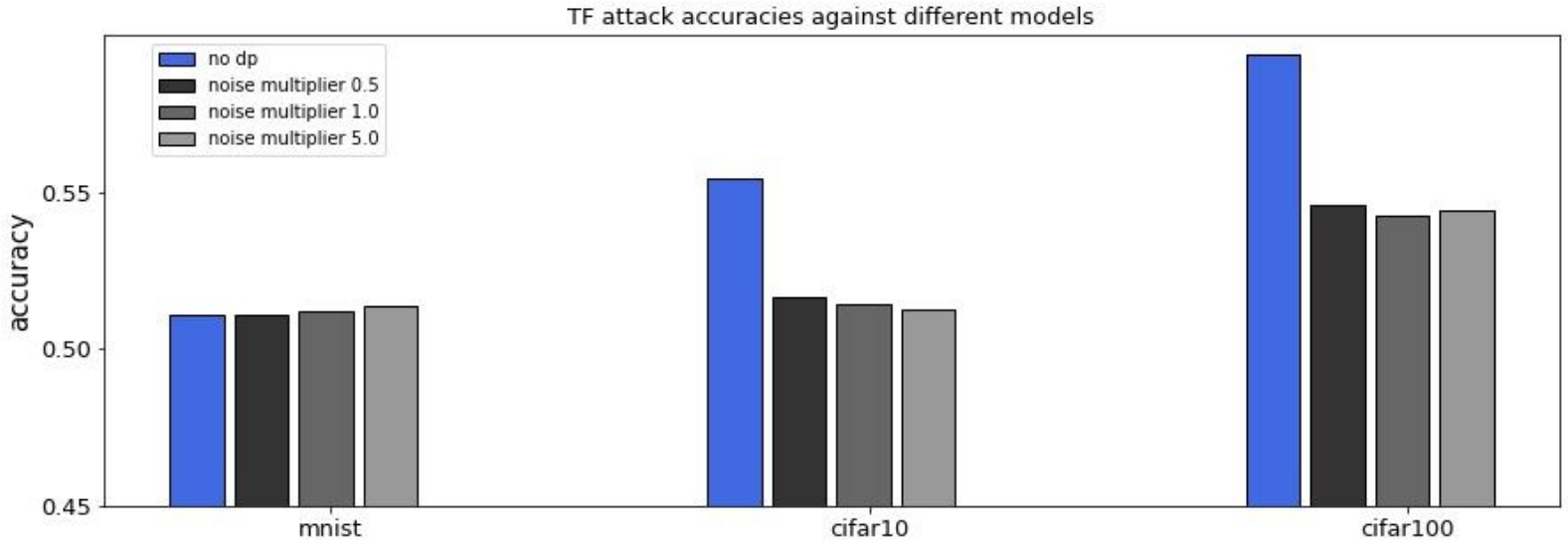
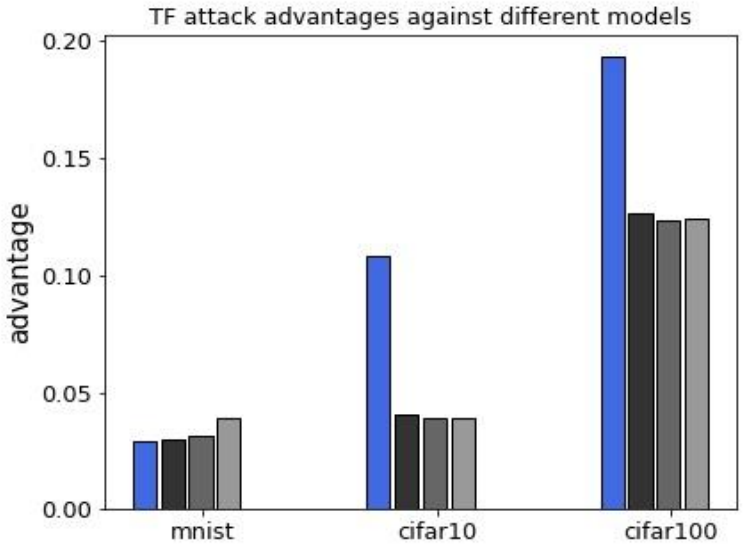


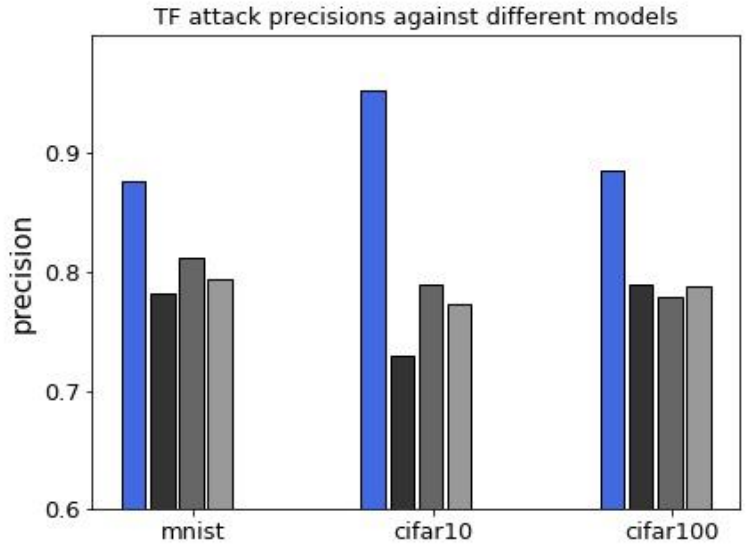
Fig. Test accuracy-epoch on CIFR100

- Privacy-utility trade off: smaller ϵ provides better privacy protection, at the cost of decreasing model performance
- DPML can alleviate overfitting, i.e. the difference between train and test accuracy

3.3 Tensorflow MIA Results



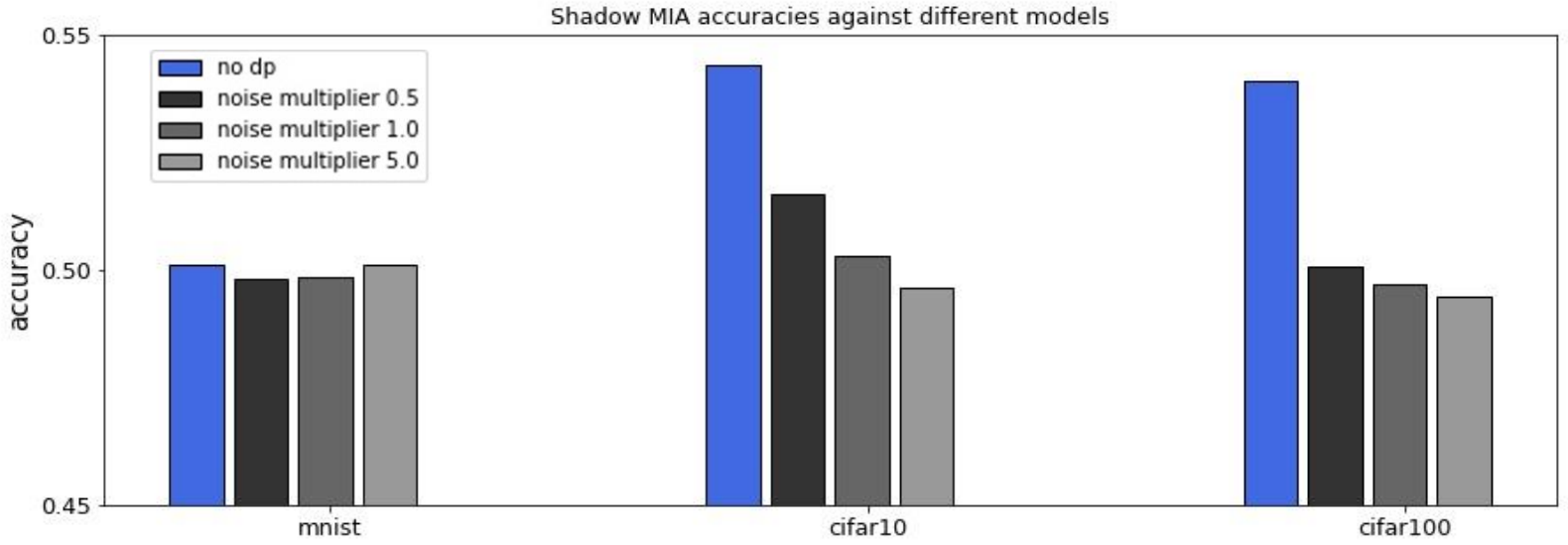


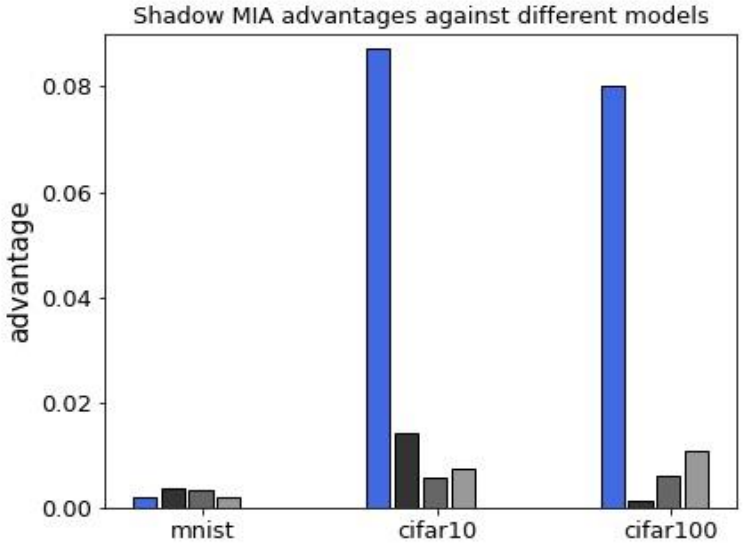


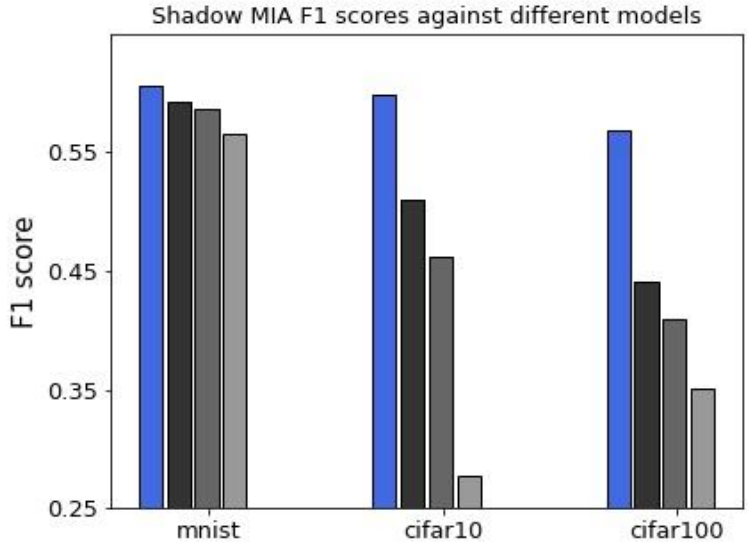
- TF MIA is a successful attack on CIFAR10 and CIFAR100
- TF MIA performance degrades on DP trained models
- Different noise levels during DPML have similar influence on TF MIA.

Note: a higher noise multiplier means smaller ϵ and stronger privacy protection

3.4 Shadow Model MIA results

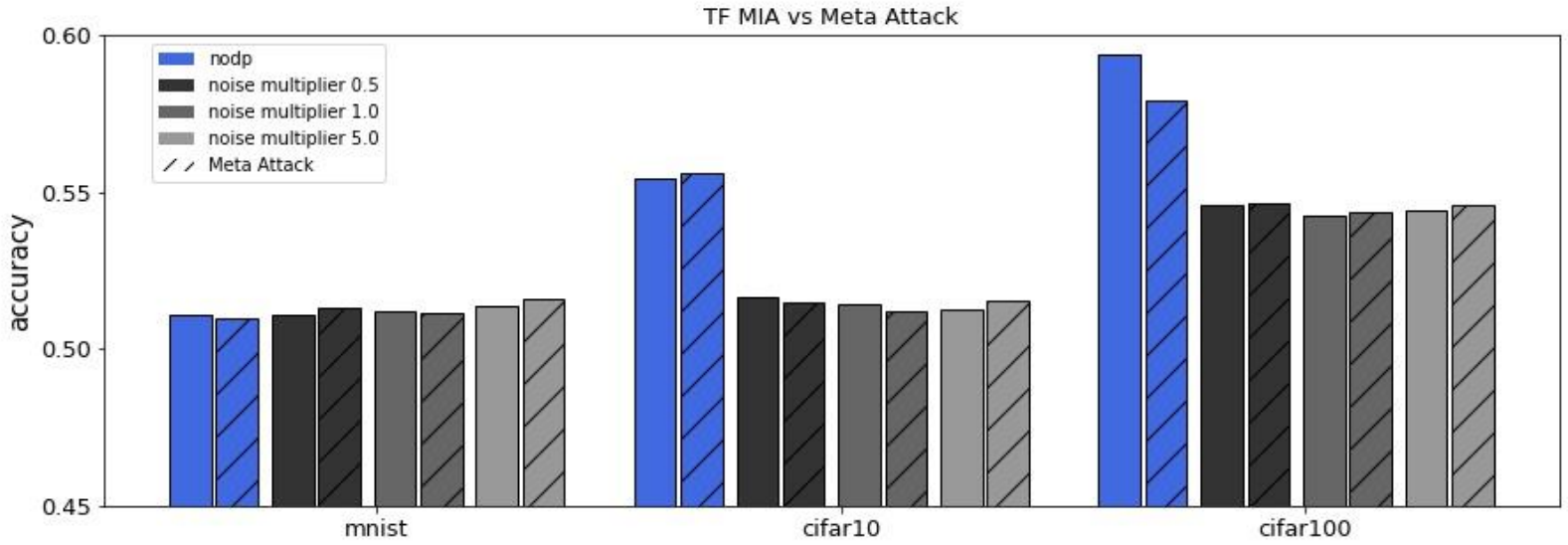


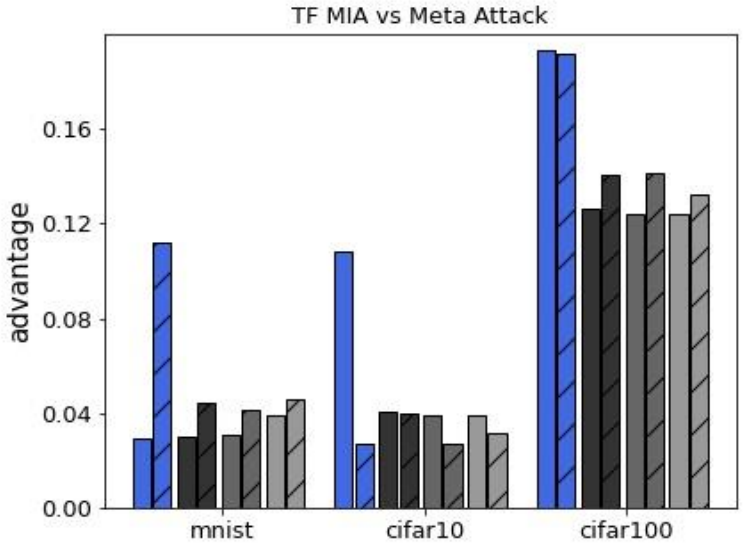


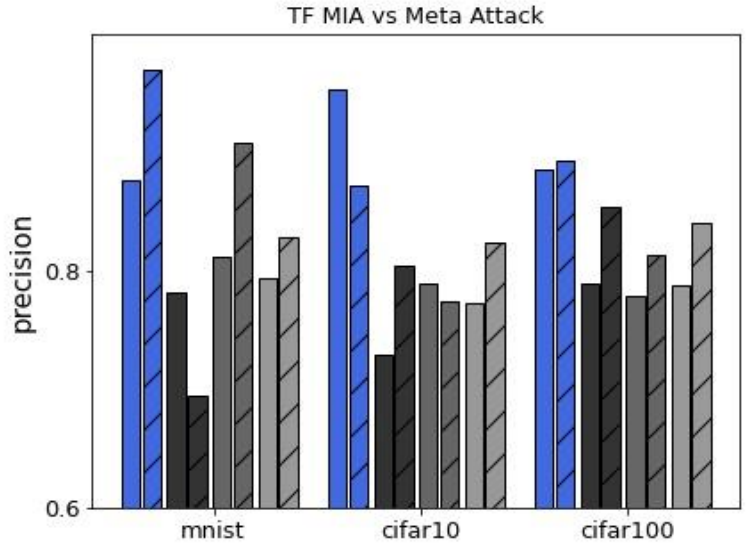


- Shadow MIA is a successful attack on CIFAR10 and CIFAR100
- Shadow MIA performance degrades on DP trained models
- Different noise levels during DPML have similar influence on TF MIA.
- Shadow MIA is similar to TF MIA
- Both shadow MIA and TF MIA does not perform well on MNIST, this is largely due to that the target model does not overfit on MNIST, and overfit on CIFAR10 and CIFAR100.

3.5 Meta Attack vs TF MIA







- Meta attack shows slightly better performance in accuracy but generally achieves distinctive improvement with respect to advantage and precision
- The current meta attack method could be improved, e.g. try different meta attack models or increase the number of TF attackers to produce larger training set for meta attacker.

3.6 Effects of overfitting

Tab. Effects of overfitting with VGG16

setting	train_acc	test_acc	Shadow MIA (acc,adv)	TF MIA (acc,adv)
CIFAR10, NO DP	0.97	0.61	(0.77, 0.36)	(0.84, 0.67)
CIFAR10, $\epsilon = 6.9$	0.78	0.73	(0.52, 0.04)	(0.59, 0.08)
CIFAR100, NO DP	0.99	0.473	(0.81, 0.52)	(0.91, 0.74)
CIFAR100, $\epsilon = 9.4$	0.64	0.55	(0.56, 0.09)	(0.67, 0.13)
MNIST, NO DP	0.99	0.75	(0.75, 0.46)	(0.78, 0.54)
MNIST, $\epsilon = 6.2$	0.91	0.89	(0.55, 0.09)	(0.61, 0.12)

Note: Overfitting is achieved by disenabling drop-out of VGG16 model and data-augmentation during training

Contents

Conclusion and Future work

- **Conclusion**:
  - Machine learning models are vulnerable to the membership inference attack and suffer from privacy risks. The privacy leakages is especially severe when the model is overfitted.
  - DP training can effectively prevent the prevent the privacy leakage from MIA and providing privacy protection to the models. However, strong DP protection (small ϵ) may lead to a decrease of model utility.
  - Empirical privacy risk analysis can shed light on the understanding of (ϵ, δ) of DP and the specific training parameters of DPML.
- **Future work**
  - Experimentation on more complicated DP training settings e.g. adaptive clipping bound, noise decay, different δ, etc.;
  - Improve meta attack (consider also methods like majority vote or other ensemble methods), experimentation on other novel privacy attacks (use model weights/intermediate outputs [4]);
  - Improve performance of MIA against non-overfitted model, explore the connection between overfitting and MIA attack

References

[1] Dwork, Cynthia. "Differential privacy: A survey of results." *International conference on theory and applications of models of computation*. Springer, Berlin, Heidelberg, 2008.

[2] Shokri, Reza, et al. "Membership inference attacks against machine learning models." *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

[3] https://github.com/tensorflow/privacy

[4] Leino, Klas, and Matt Fredrikson. "Stolen memories: Leveraging model memorization for calibrated white-box membership inference." *USENIX Security,* 2020.

[5] Yeom, Samuel, et al. "Privacy risk in machine learning: Analyzing the connection to overfitting." *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018.

[6] Abadi, Martin, et al. "Deep learning with differential privacy." Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016.