

MEMBERSHIP INFERENCE ATTACKS AGAINST MACHINE LEARNING MODELS

Presented By: Reza Shokri; Marco Stronati ; Congzheng Song ; Vitaly Shmatikov
(Cornell University)



(38th IEEE Symposium on Security and Privacy)
May 2017

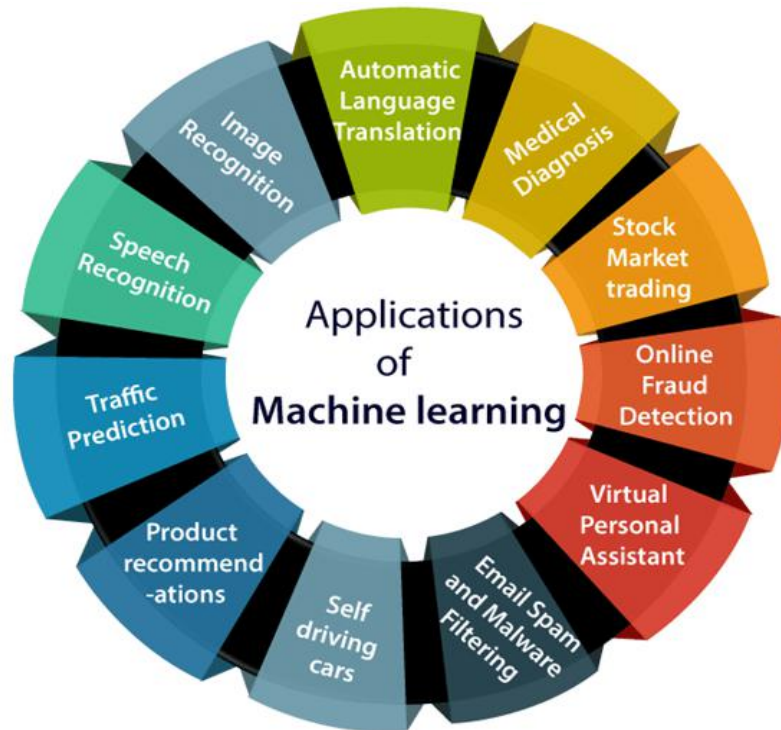
Richmond Asiedu Agyapong
04/21/2020

**Machine Learning For Cyber
Security**
Instructor:
Dr. Mahmoud Nabil Mahmoud

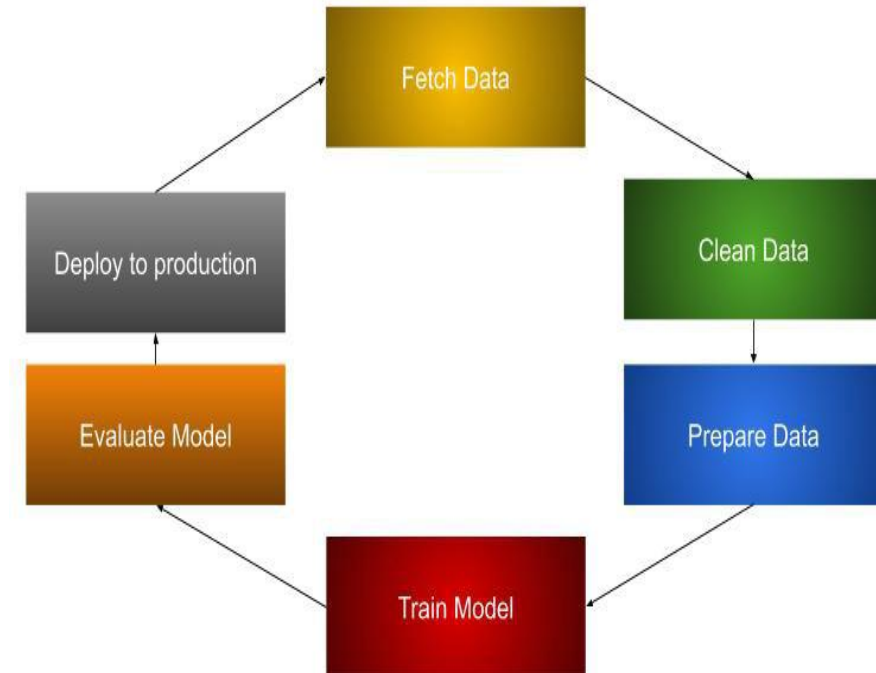
Outline

- **Machine Learning / MLAAS**
- **Challenges (Privacy)**
- **Privacy Attacks**
- **Membership Inference Attacks**
 - > **Overview**
 - > **Building Attack Model**
- **Evaluation of MIA**
- **Code / Implementation**
- **Mitigation**
 - > **Strategies**
 - > **Evaluation**

Machine Learning

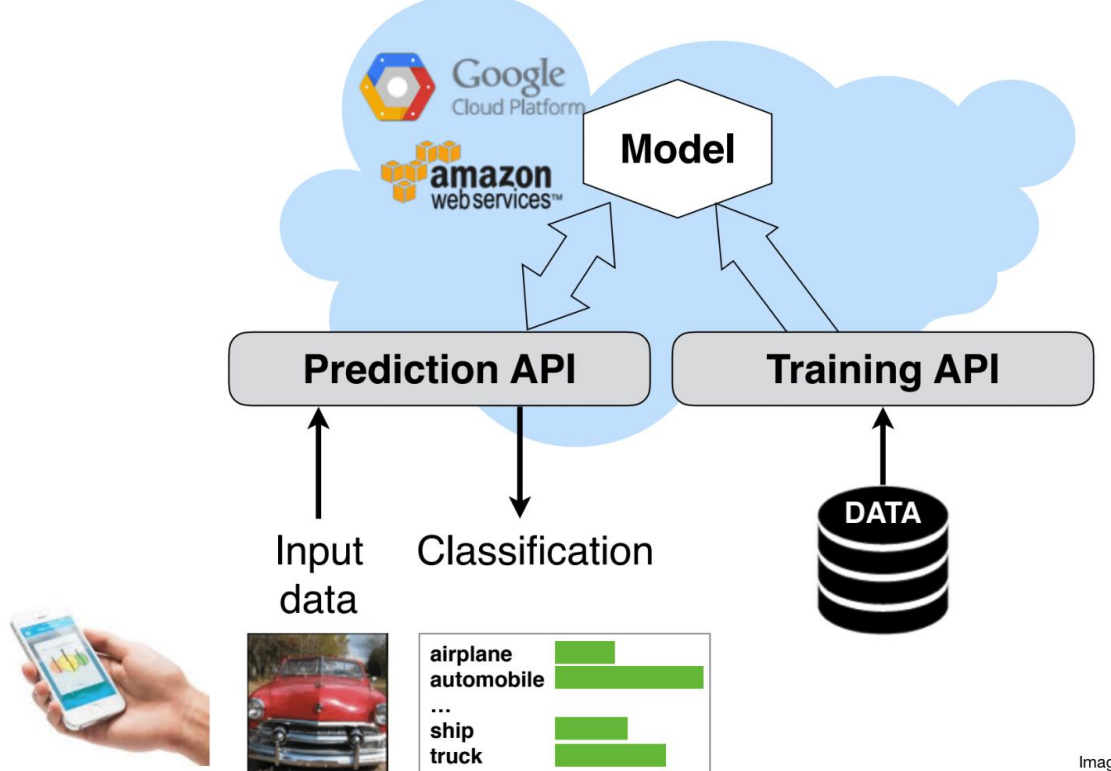


Plethora of a applications



Data is essential

Machine Learning as a Service

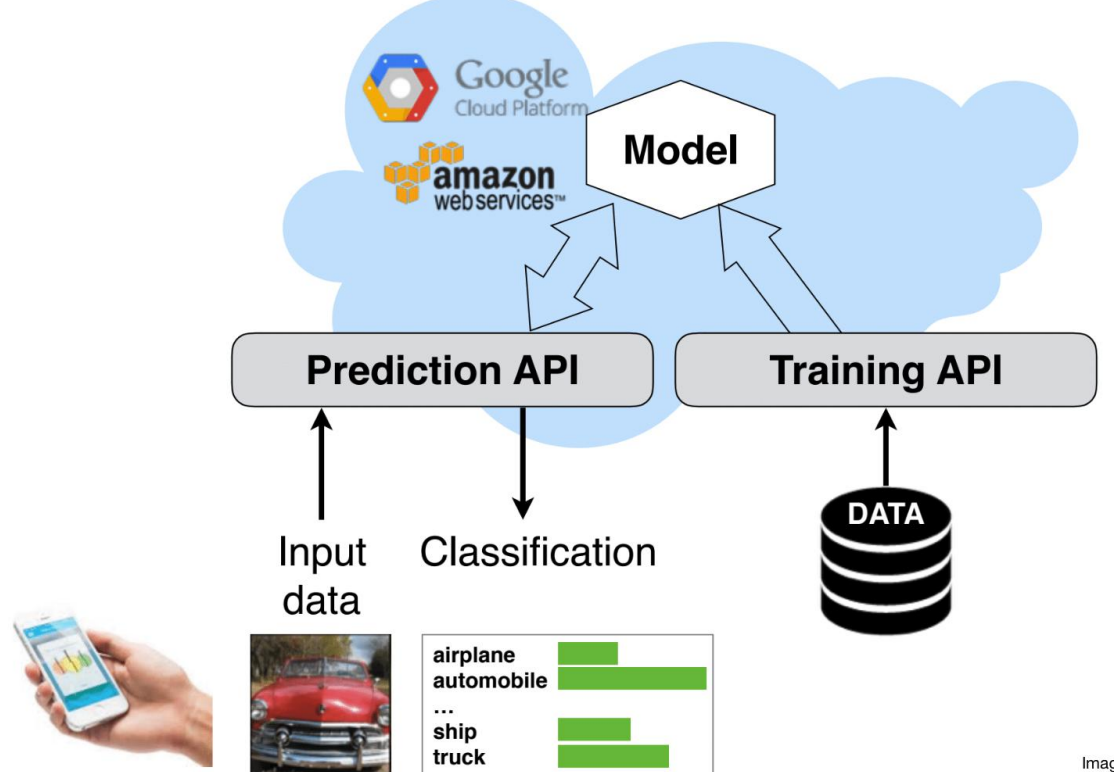


Privacy: Greatest Challenge

Membership Inference Attacks:

Determine whether a data record was used in target's training

Machine Learning as a Service



Privacy: Greatest Challenge

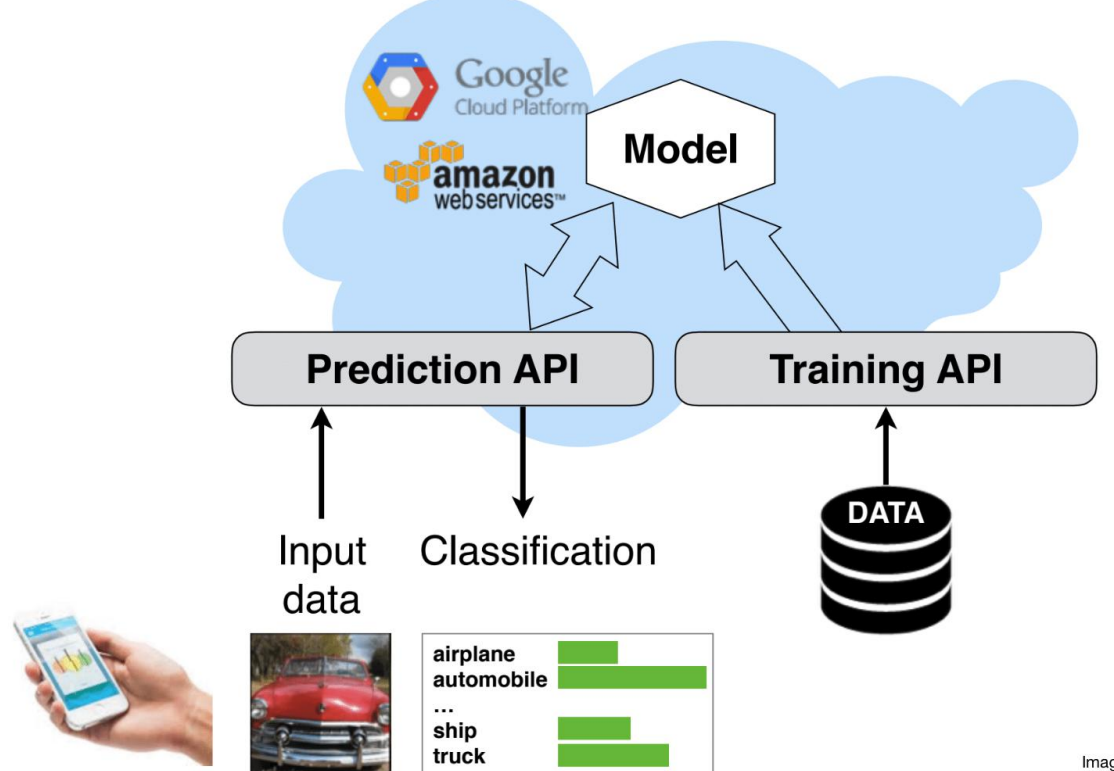
Membership Inference Attacks:

Determine whether a data record was used in target's training

Attribute Inference Attacks:

Determine facts about data that are otherwise hidden and private.

Machine Learning as a Service



Privacy: Greatest Challenge

Membership Inference Attacks:

Determine whether a data record was used in target's training

Attribute Inference Attacks:

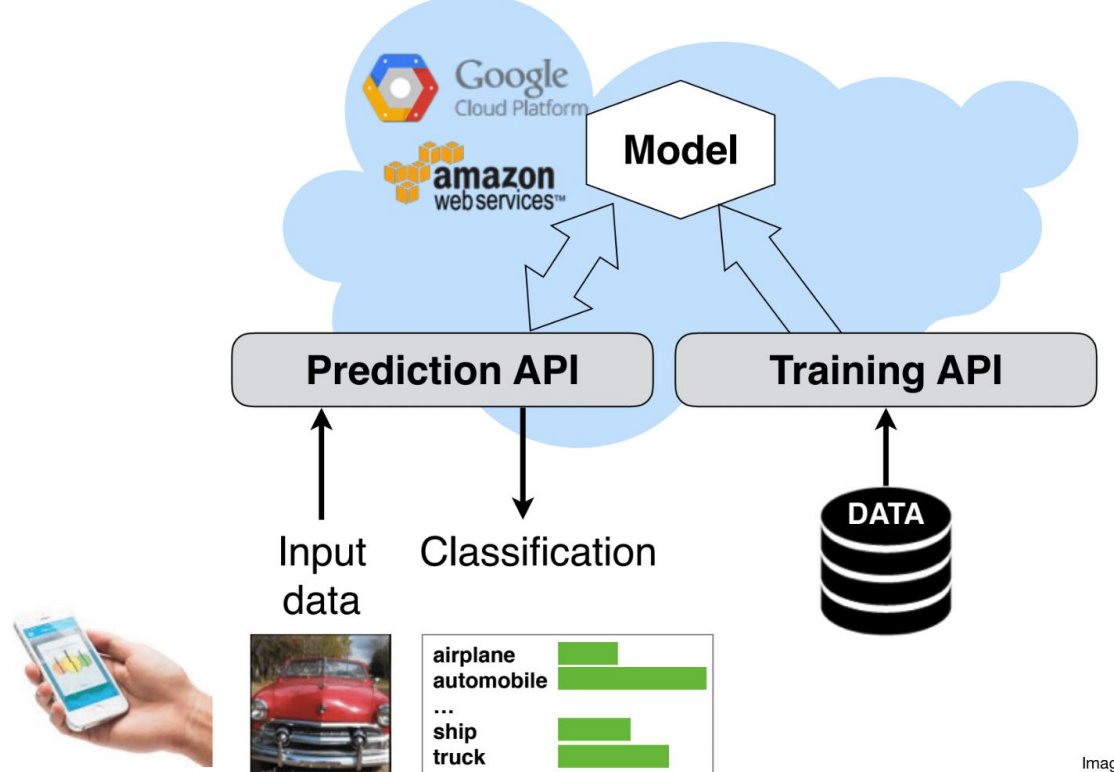
Determine facts about data that are otherwise hidden and private.

Model Inversion Attacks:

Identify features that characterize a class/ an input.

Model Extraction Attacks:

Machine Learning as a Service



Privacy: Greatest Challenge

Membership Inference Attacks:

Determine whether a data record was used in target's training

Attribute Inference Attacks:

Determine facts about data that are otherwise hidden and private.

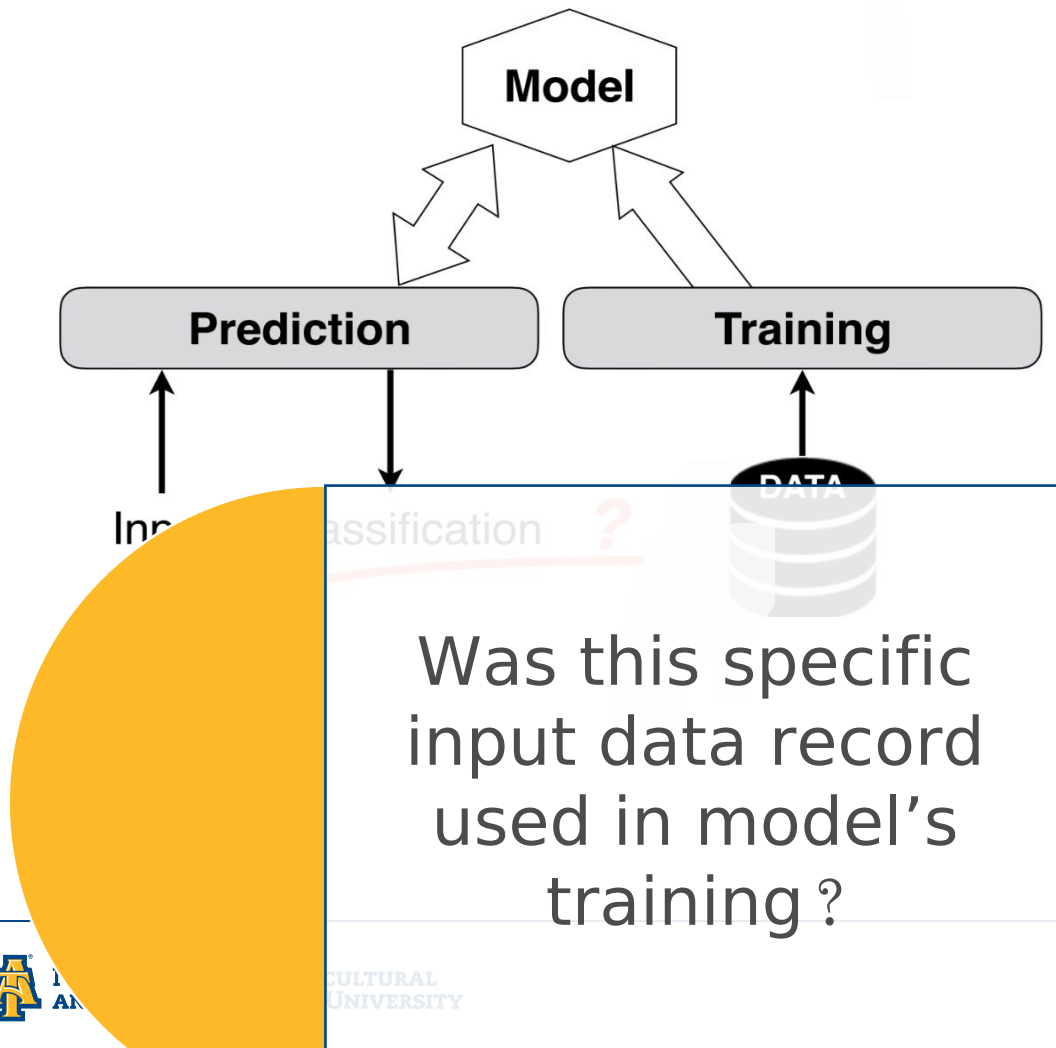
Model Inversion Attacks:

Identify features that characterize a class/ an input.

Model Extraction Attacks:

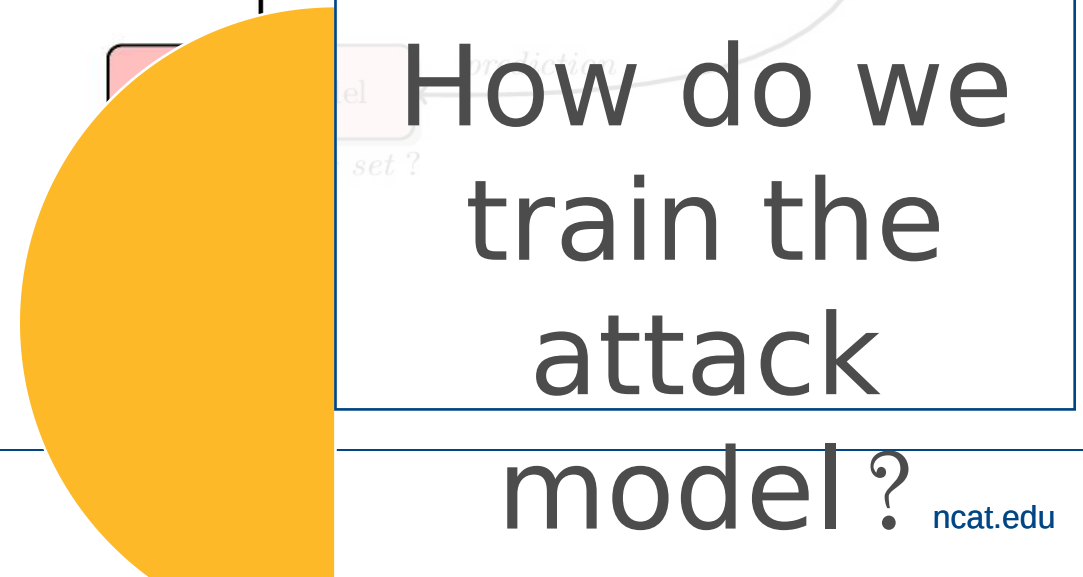
extract the parameters of a model to construct a model whose predictive performance on validation data is similar to the target model.

Membership Inference Attack



- Build a binary classifier (Attack Model)

- Take target's predictions as inputs



Building Attack Model


- Learn behavior of target prediction
 - > (i.e. difference b/n Train set and Test/other set)

- Overfitting



- Training data = predictions

Shadow Models

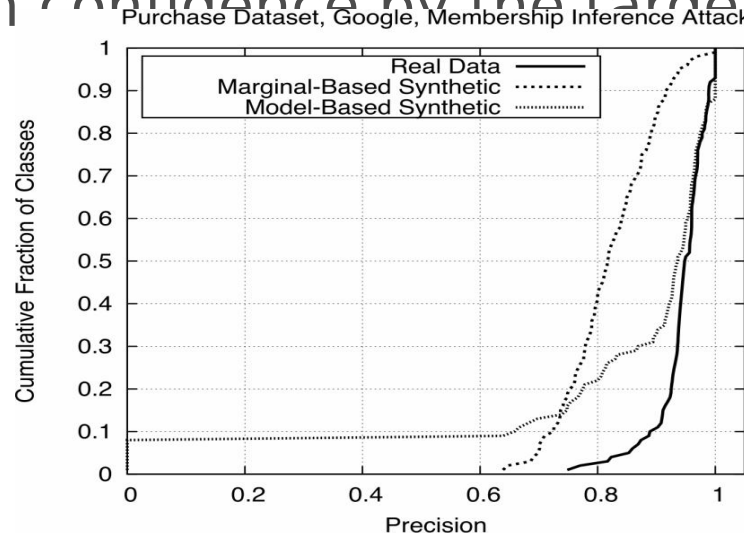
- Similarities to Target Model
 - > type and architecture
 - > training Dat 



Shadow Model \approx Target Model

Obtaining Data for Shadow Models

- **Noisy real data:** similar to training data of target model (i.e. drawn from same distribution)
- **Synthetic data:** use a sampling algorithm to obtain data classified with high confidence by the target model

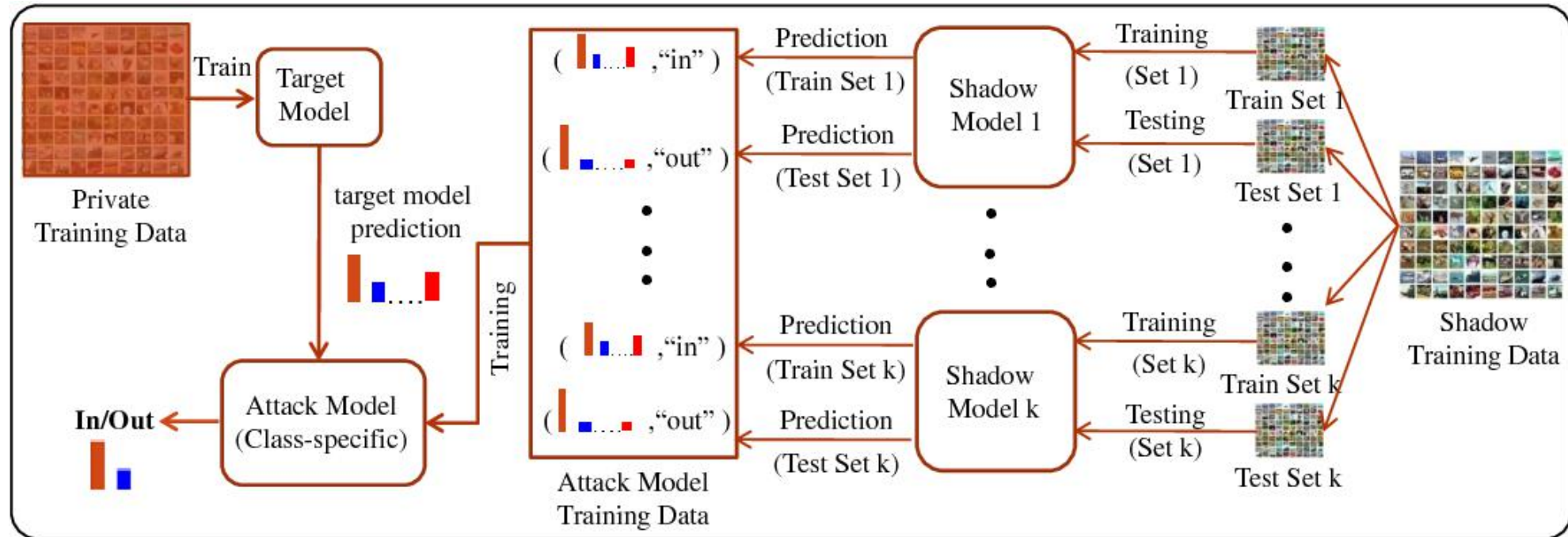


```

1: procedure SYNTHESIZE(class : c)
2:    $x \leftarrow \text{RANDRECORD}()$   $\triangleright$  initialize a record randomly
3:    $y_c^* \leftarrow 0$ 
4:    $j \leftarrow 0$ 
5:    $k \leftarrow k_{\max}$ 
6:   for iteration = 1  $\dots$  itermax do
7:      $y \leftarrow f_{\text{target}}(x)$   $\triangleright$  query the target model
8:     if  $y_c \geq y_c^*$  then  $\triangleright$  accept the record
9:       if  $y_c > \text{conf}_{\min}$  and  $c = \arg \max(y)$  then
10:        if rand() <  $y_c$  then  $\triangleright$  sample
11:          return x  $\triangleright$  synthetic data
12:        end if
13:      end if
14:       $x^* \leftarrow x$ 
15:       $y_c^* \leftarrow y_c$ 
16:       $j \leftarrow 0$ 
17:    else
18:       $j \leftarrow j + 1$ 
19:      if  $j > \text{rej}_{\max}$  then  $\triangleright$  many consecutive rejects
20:         $k \leftarrow \max(k_{\min}, \lceil k/2 \rceil)$ 
21:         $j \leftarrow 0$ 
22:      end if
23:    end if
24:     $x \leftarrow \text{RANDRECORD}(x^*, k)$   $\triangleright$  randomize k features
25:  end for
26:  return  $\perp$   $\triangleright$  failed to synthesize
27: end procedure

```

Membership inference Attack



- Attack model is a collection of models.
- One for each output class of target model

Evaluation

- **Data**

- > CIFAR – 10, CIFAR – 100
- > MNIST
- > Purchases
- > Texas Hospital Stays
- > UCI Adult (Census Income)

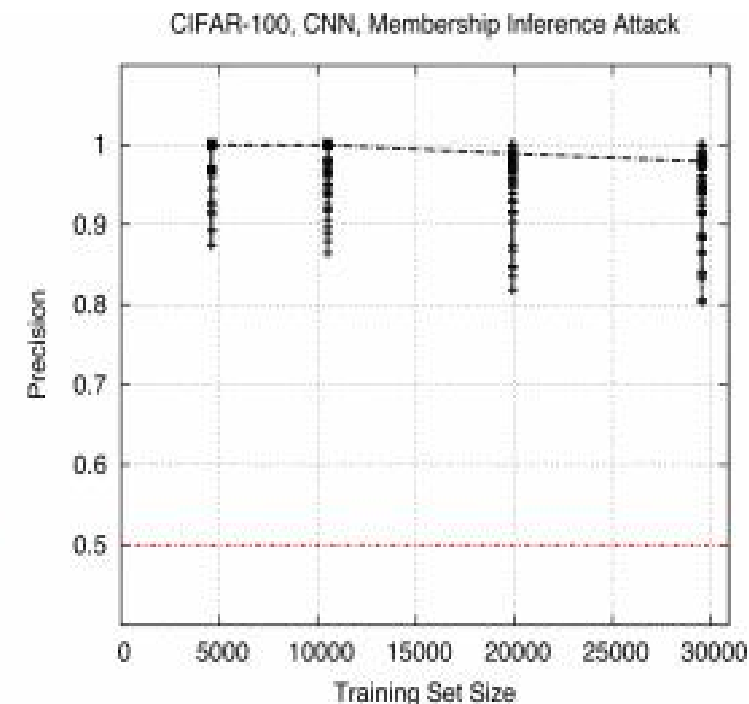
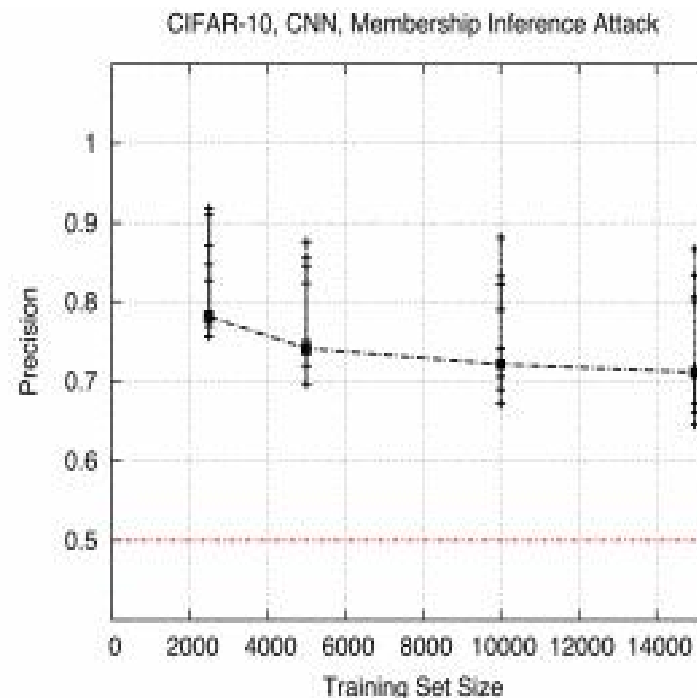
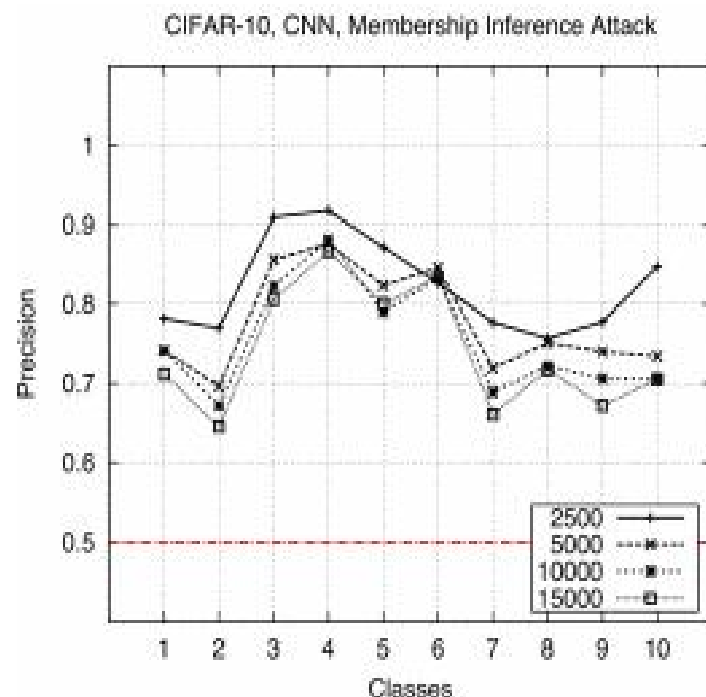
- **Target Models**

- > Amazon ML
- > Google Prediction API
- > Neural Networks (Locally trained)

- **Experimental Setup**

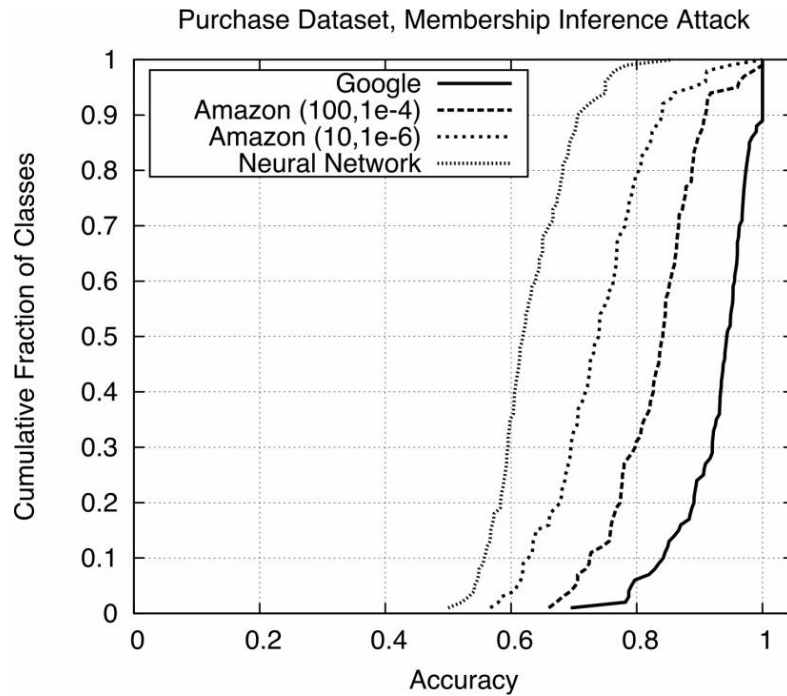
- > Run data on all models
- > CIFAR datasets run locally
- > Vary size of training dataset
- > Vary the number of shadow models

Accuracy

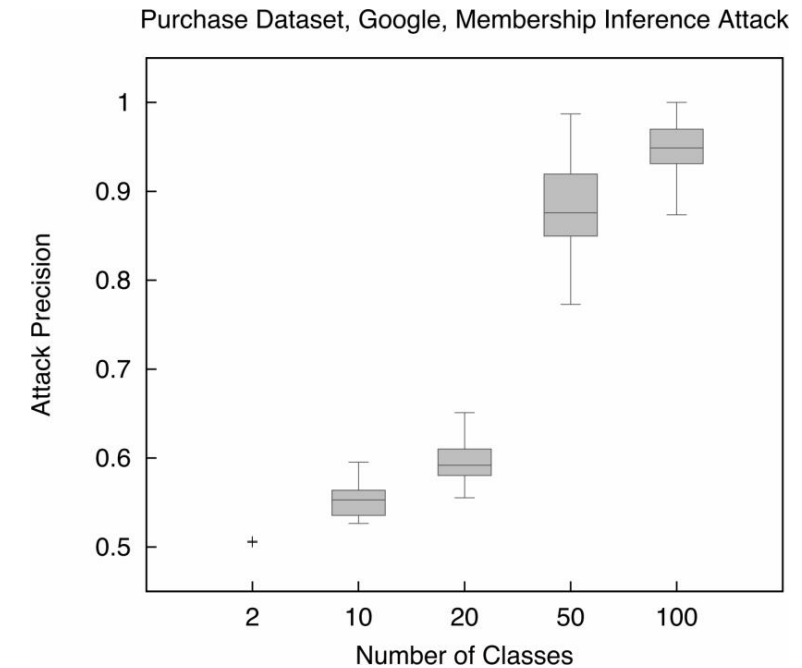


Precision of the membership inference attack against neural networks trained on CIFAR datasets. The graphs show precision for different classes while varying the size of the training datasets. The median values are connected across different training set sizes. The median precision (from the smallest dataset size to largest) is 0.78, 0.74, 0.72, 0.71 for CIFAR-10 and 1,1, 0.98,0.97 for CIFAR-100. Recall is almost 1 for both datasets. The figure on the left shows the per-class precision (for CIFAR-10). Random guessing accuracy is 0.5.

Accuracy



ML Platform	Train	Test
Google	0.999	0.656
Amazon(10, 1e-6)	0.941	0.468
Amazon(100, 1e-4)	1.0	0.504
Neural Net	0.830	0.670



Code

Library

```
from mia.estimators import ShadowModelBundle, AttackModelBundle,
prepare_attack_data
```

Build shadow model and generate data for training attack model

```
smb = ShadowModelBundle(
    target_model_fn,
    shadow_dataset_size=SHADOW_DATASET_SIZE,
    num_models=FLAGS.num_shadows,
)
X_shadow, y_shadow = smb.fit_transform(att_X_train, att_y_train)
```

Code to generate Attack Models

```
amb = AttackModelBundle(attack_model_fn, num_classes=NUM_CLASSES)
amb.fit(X_shadow, y_shadow)
```

Prepare data for Attack

```
attack_test_data, real_membership_labels = prepare_attack_data(
    target_model, data_in, data_out)

attack_guesses = amb.predict(attack_test_data)
attack_accuracy = np.mean(attack_guesses == real_membership_labels)
```


Code / Implementation

Mitigation

Strategies

- Restrict the Prediction Vector to Top k Classes
- Coarsen Precision of the Prediction Vector
- Increase Entropy of the Prediction Vector
- Use Regularization

Evaluation of Strategies

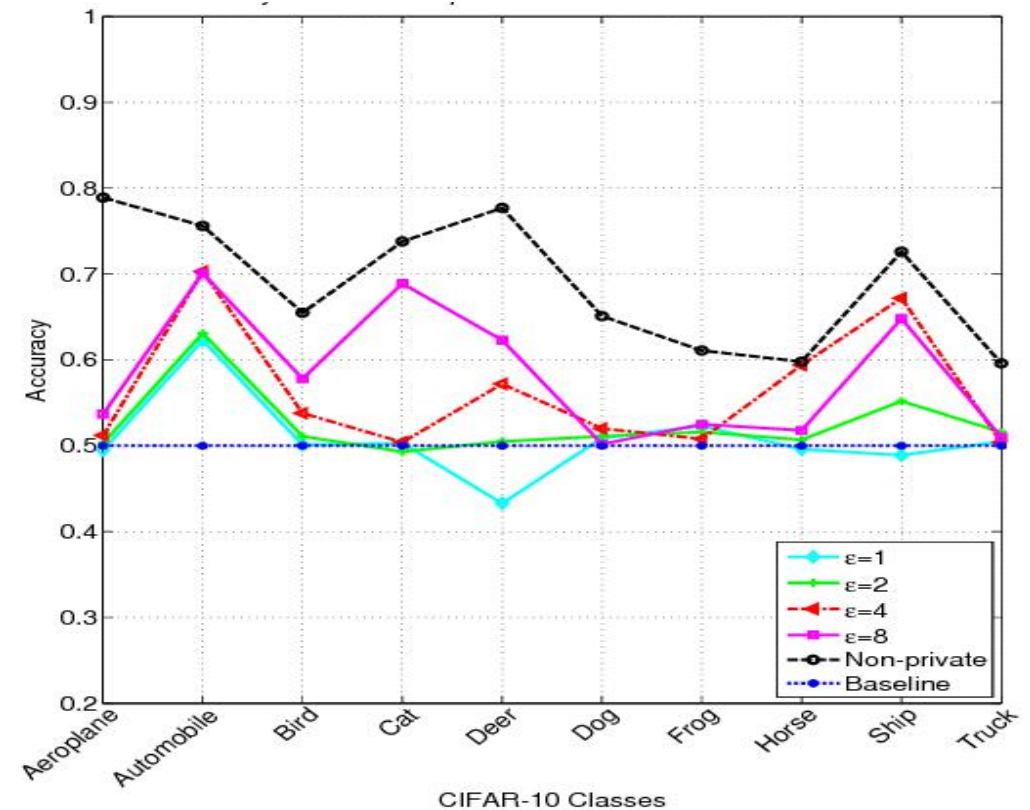
Purchase dataset	Testing Accuracy	Attack Total Accuracy	Attack Precision	Attack Recall
No Mitigation	0.66	0.92	0.87	1.00
Top $k = 3$	0.66	0.92	0.87	0.99
Top $k = 1$	0.66	0.89	0.83	1.00
Top $k = 1$ label	0.66	0.66	0.60	0.99
Rounding $d = 3$	0.66	0.92	0.87	0.99
Rounding $d = 1$	0.66	0.89	0.83	1.00
Temperature $t = 5$	0.66	0.88	0.86	0.93
Temperature $t = 20$	0.66	0.84	0.83	0.86
L2 $\lambda = 1e - 4$	0.68	0.87	0.81	0.96
L2 $\lambda = 1e - 3$	0.72	0.77	0.73	0.86
L2 $\lambda = 1e - 2$	0.63	0.53	0.54	0.52

Differential privacy

Strategy

Evaluation

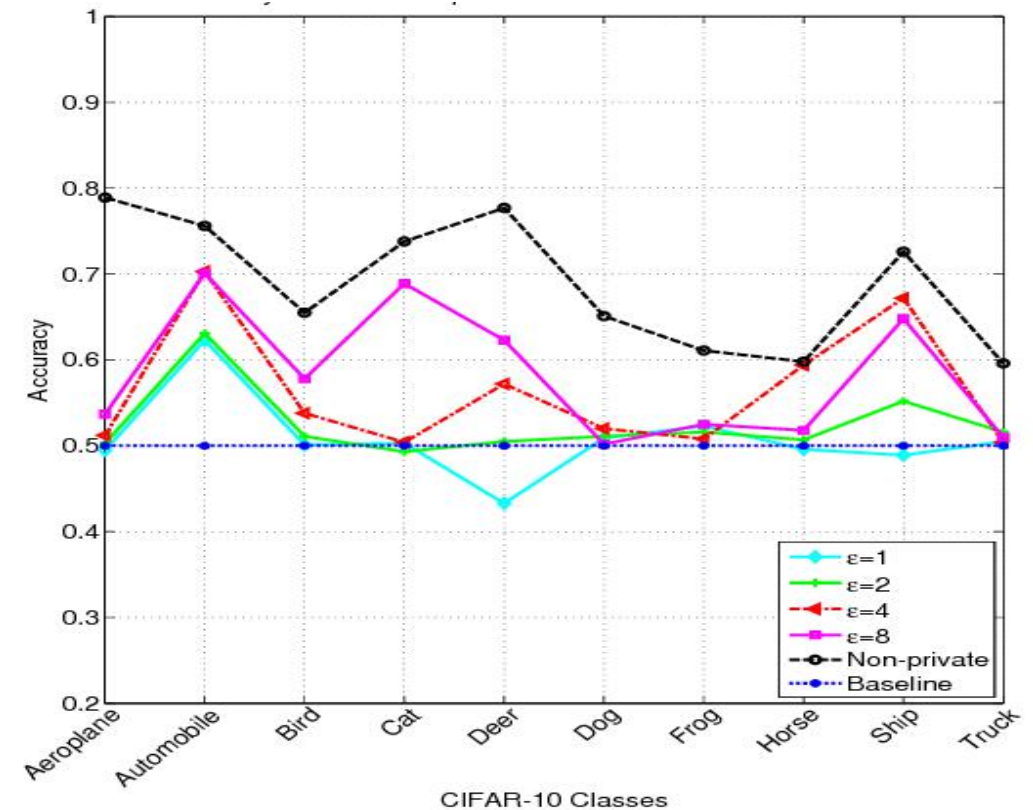
- Use Differential privacy in training/ building Target model
- (ϵ, δ) Differential privacy
- The distribution of output $M(D)$ is nearly the same as $M(D')$
- D & D' differ slightly
- ϵ is info leakage
- δ is small probability of failure.



Differential privacy Evaluation (Cnt'd)

Datasets	$\epsilon=1$	$\epsilon=2$	$\epsilon=4$	$\epsilon=8$	non-private
CIFAR-10 (train)	0.247	0.450	0.608	0.686	0.944
CIFAR-10 (test)	0.253	0.450	0.607	0.681	0.737
MNIST (train)	0.762	0.874	0.909	0.937	0.999
MNIST (test)	0.757	0.870	0.906	0.932	0.970

- Poor utility – privacy trade off



Adversarial Examples Strategy

- Add carefully crafted noise to confidence vector of Target Model
- Provide adversarial input to Attack Model
- Better utility – privacy trade off

