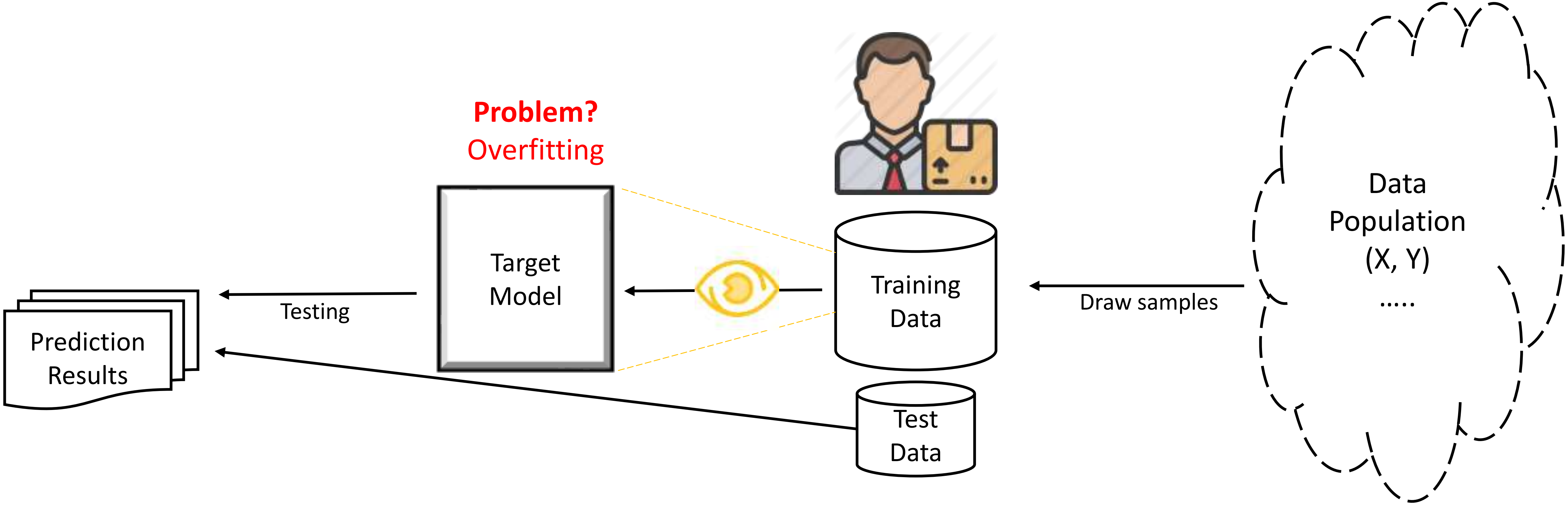
Membership Inference Attacks Against Machine Learning Models

Reza Shokri, Marco Stronati, Congzheng Song and Vitaly Shmatikov

2017 IEEE Symposium on Security and Privacy

1. Background

Supervised Learning

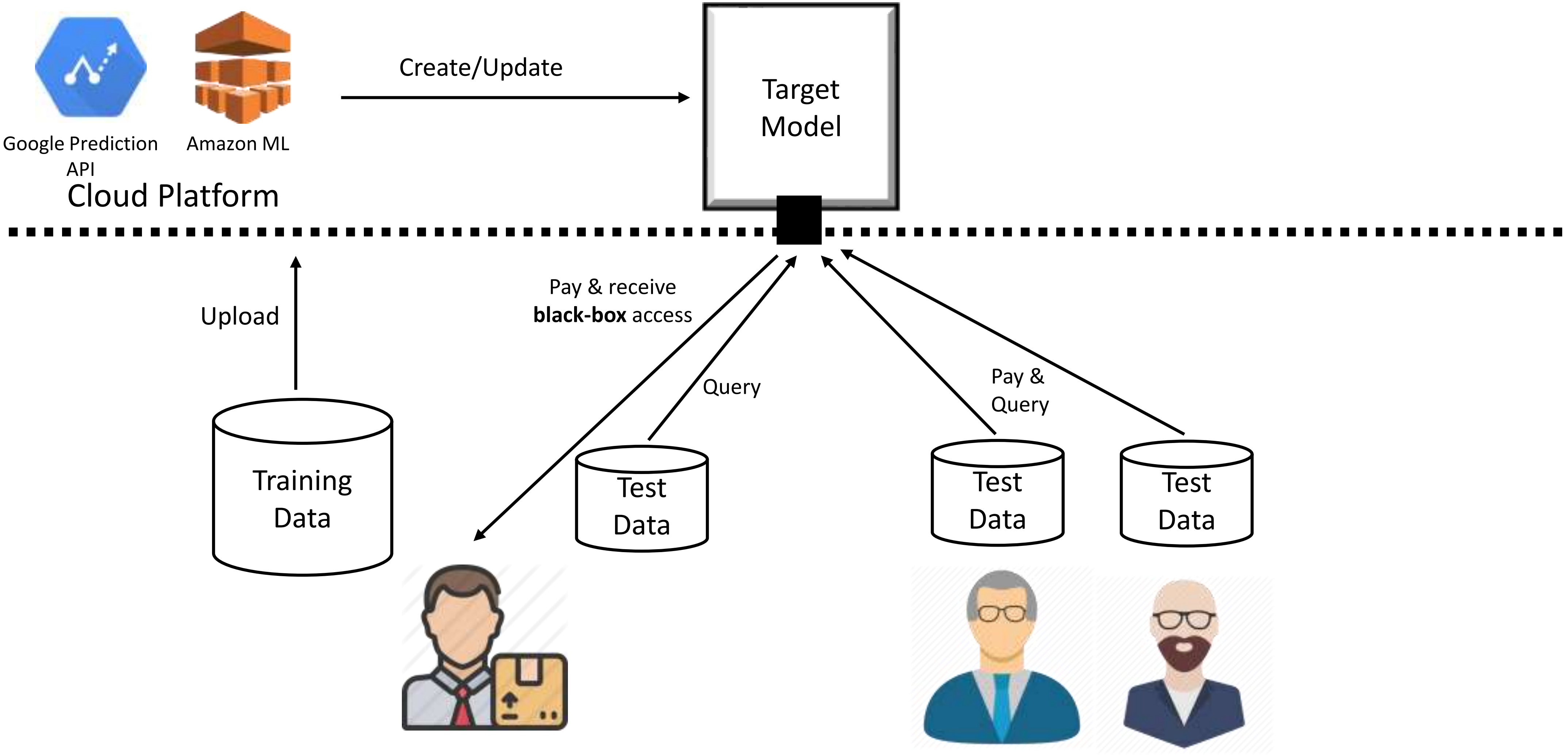


GOAL : Perform a Classification Task

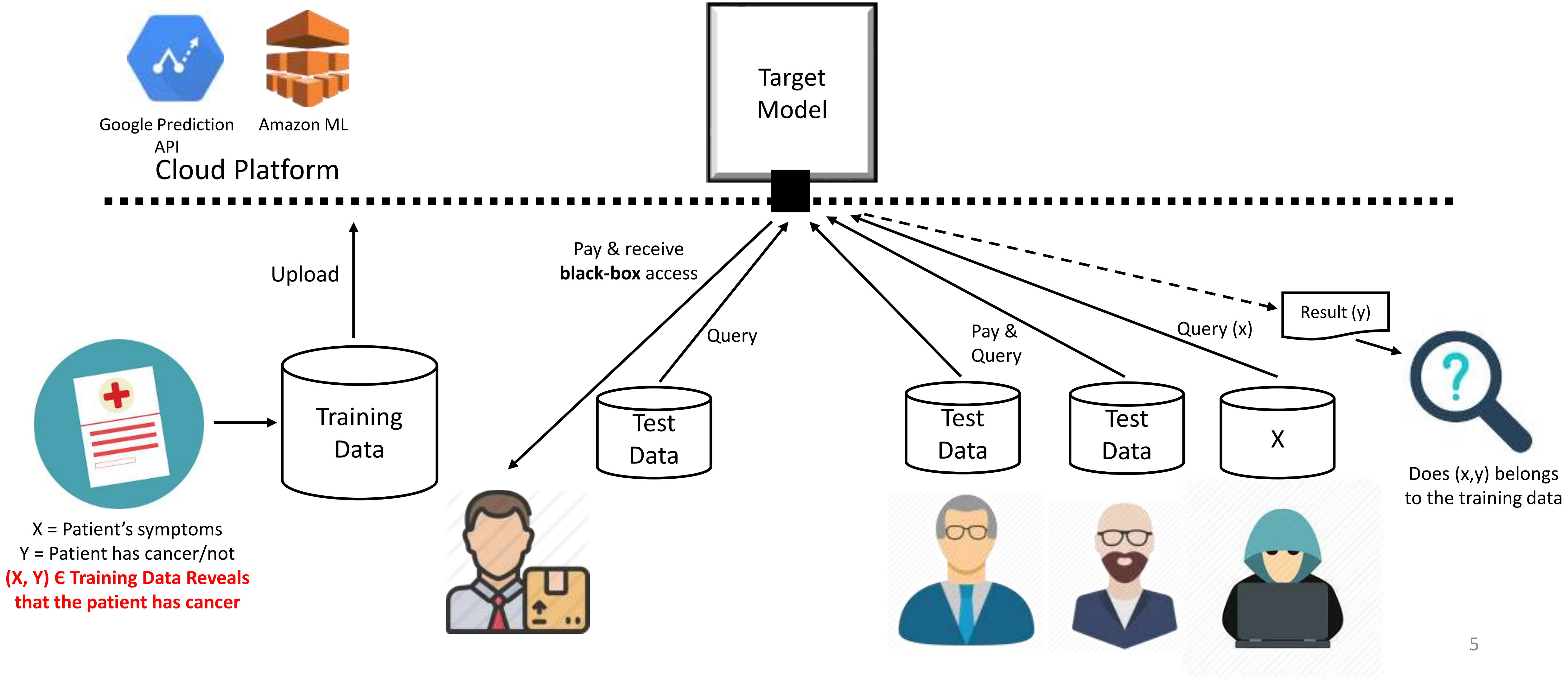
Learn f: X -> Y

Learn relationship between X & Y

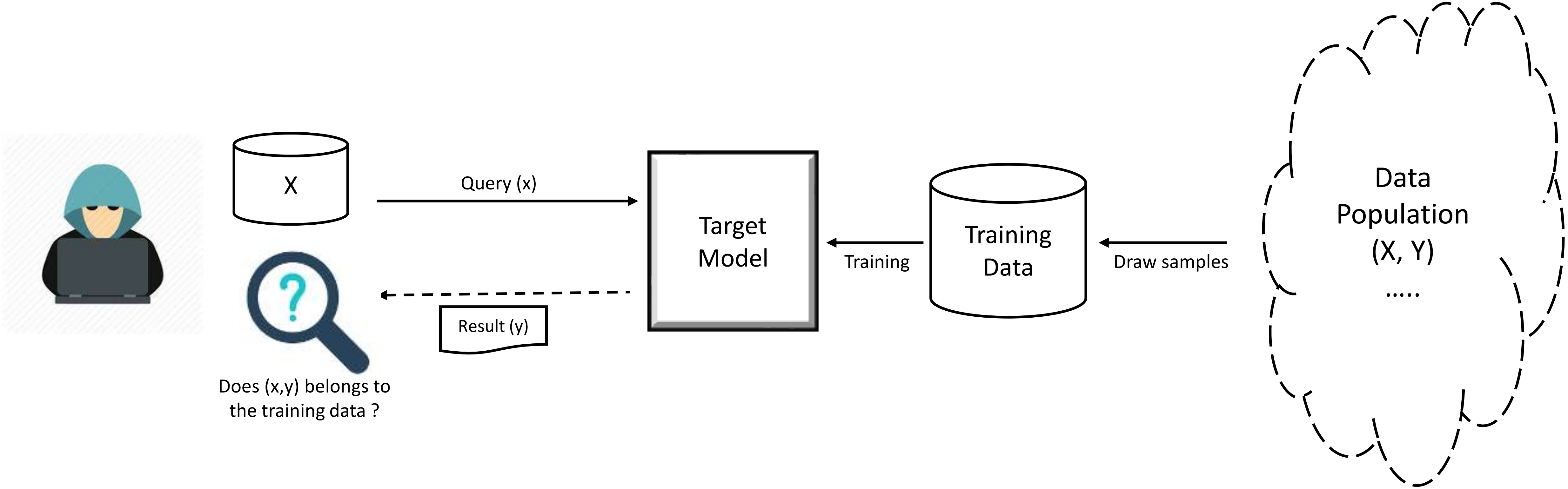
Machine Learning as a Service



Privacy Breach in ML as a Service

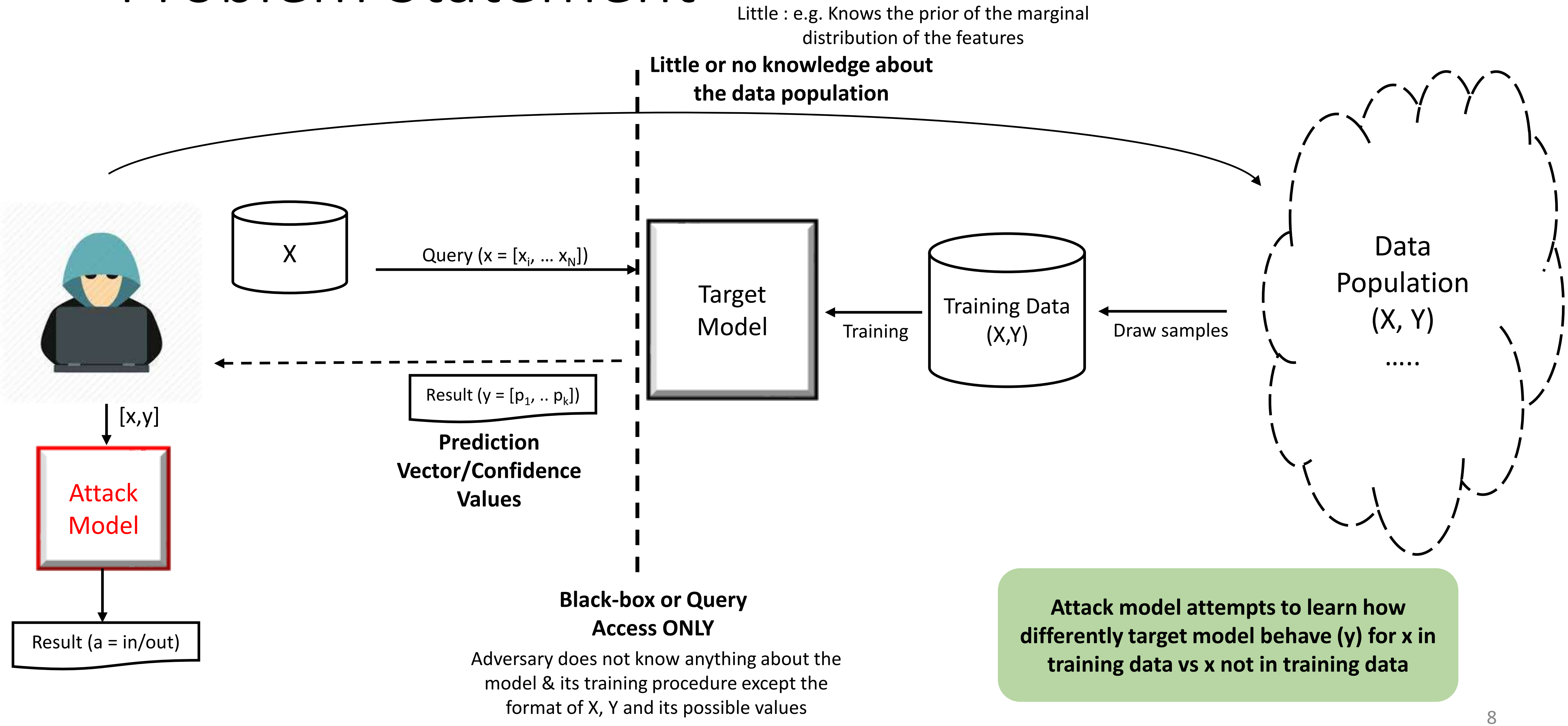


Membership Inference Attack

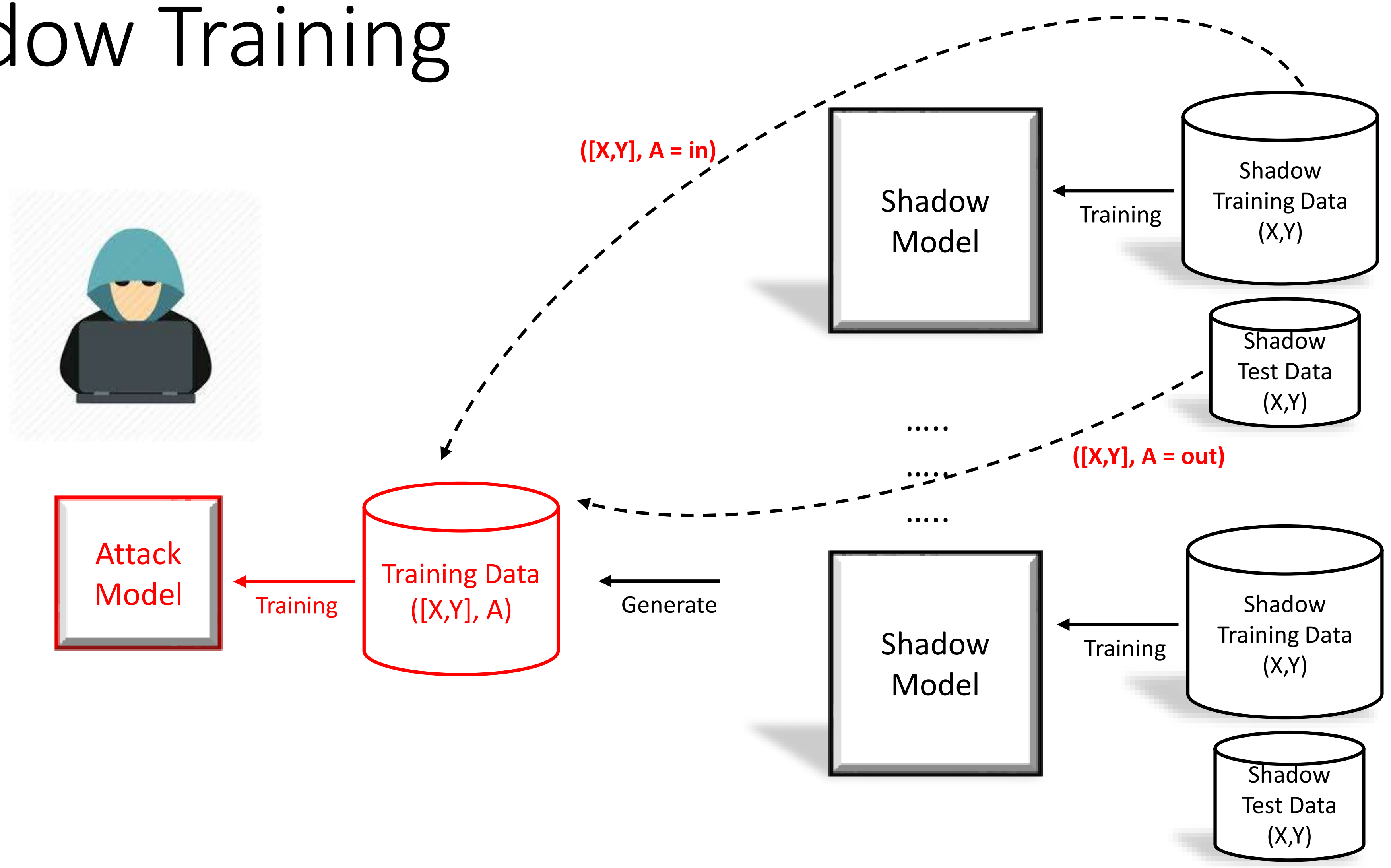


2. Problem & Solution

Problem Statement



Shadow Training



Imitate the Behavior of Target Model

Generate Shadow Train & Test Data

Knowledge of the Adversary
(about data population)

Model-based synthesis

Knowledge: Nothing

Method: Generate data using target model itself

Statistics-based synthesis

Knowledge: prior of the distribution of features of X

Method: Sample data from distribution

Noisy real data

Knowledge: Noisy version of the training data

Method: Flip binary values of 10-20% features in X

Model-based Synthesis

For each class k:

x <- randomly generate

iterate till you find enough amount of shadow data:

y <- $f_{target}(x)$

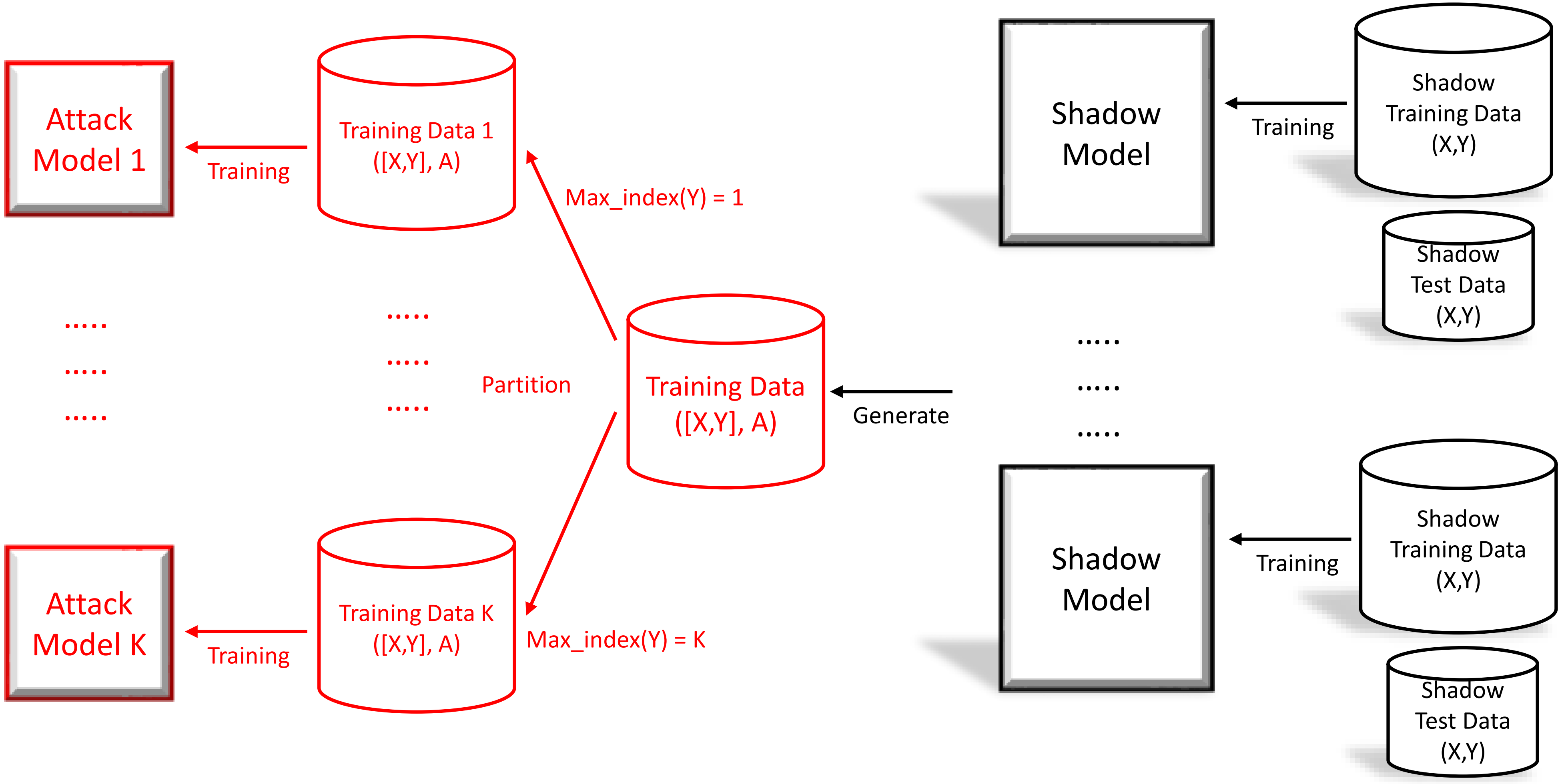
if y_k > confidence threshold:

sample x

x <- randomly generate by flipping few constant number of random features

X which are classified with high confidence (y_c) by target model should be similar to the training data

Training Attack Model



During testing [x,y] is applied to Max_index(y)th attack model

3. Evaluation

Data

Range of class size is explored (binary to 100)

Dataset	CIFAR	Purchase	Location	Hospital Stay	MNIST	UCI Adult
X	32*32 size images	600 features	446 binary features	6170 binary features	32*32 images	14
K = |Y|	10, 100	2, 10, 20, 50 & 100	30	100	10	2
Target Model	Neural Network (NN)	Google P-API, Amazon ML, NN	Google P-API	Google P-API	Google P-API/ Amazon ML	Google P-API/ Amazon ML
Number of Shadow Models	100	20	60	10	50	20

Trained only using a NN

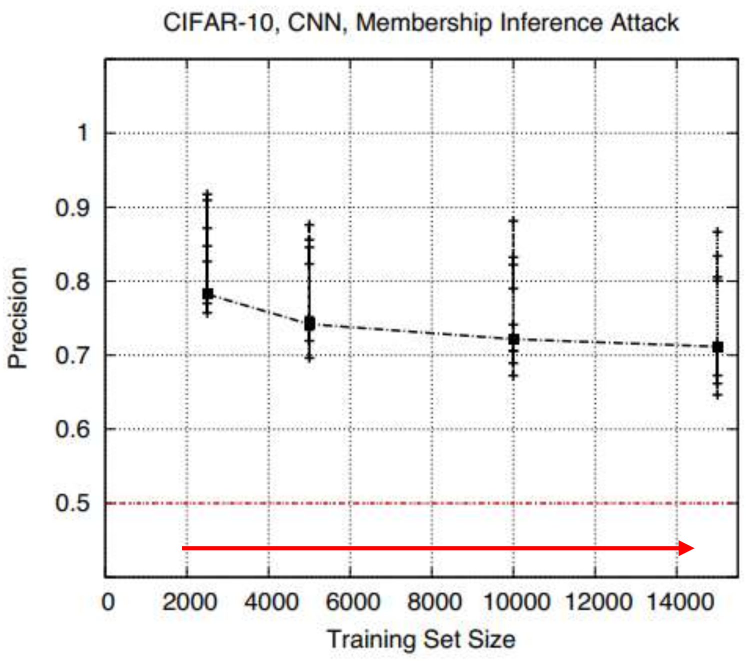
Google P-API : no control of the training
Amazon ML : # of epochs & regularization amount are changed
Two set of configurations of Amazon ML: (10, 1e-6) & (100, 1e-4)

Evaluation Setup

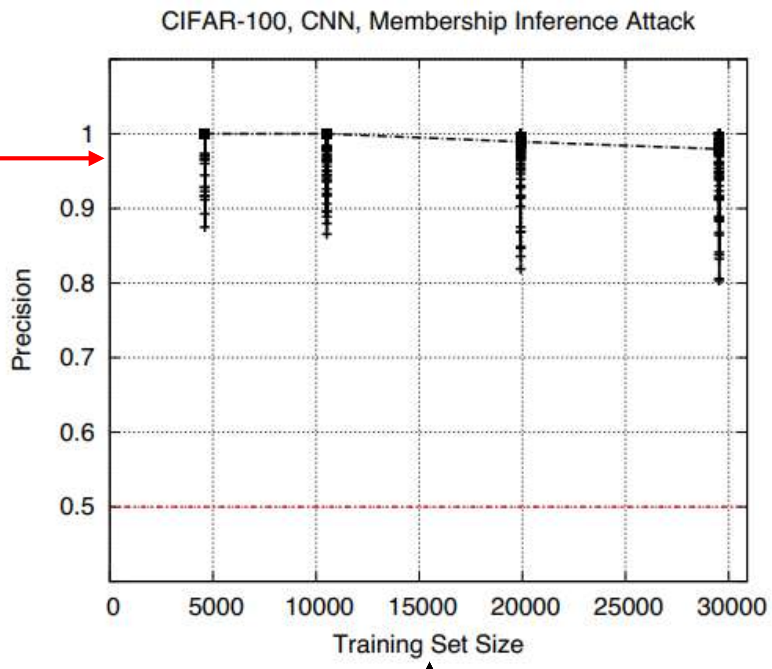
- Metrices
    1. Precision – Fraction of records inferred by attack model as members of the training dataset that are indeed members
    2. Recall – Fraction of members that are correctly identified as members by the attack model
- Test set : 50 % members & 50 % non-members of target model

    Baseline precision = 0.5
- Attack models are trained using similar architecture (NN/ Google P-API)

[R1] Number of Classes & Training Set Size



K = 10



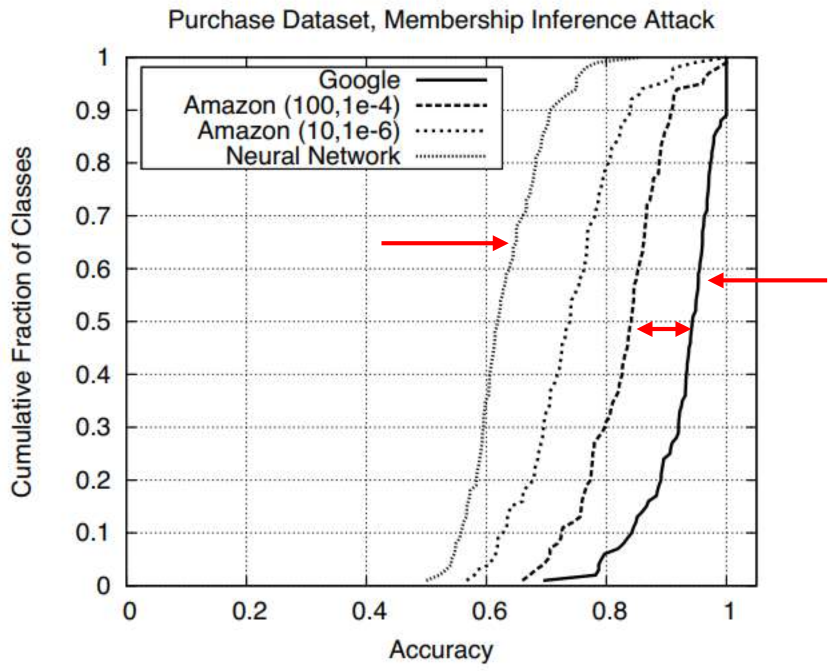
K = 100

[O1] As training data size target model increases, attack model's precision drops

[O2] As K increases, information leakage is high (precision close to 1)

[R2] Overfitting & Model Types

ML Platform	*Training*	*Test*	Overfitting
Google	0.999	0.656	0.34
Amazon (10,1e-6)	0.941	0.468	0.5
Amazon (100,1e-4)	1.00	0.504	0.5
Neural network	0.830	0.670	0.16

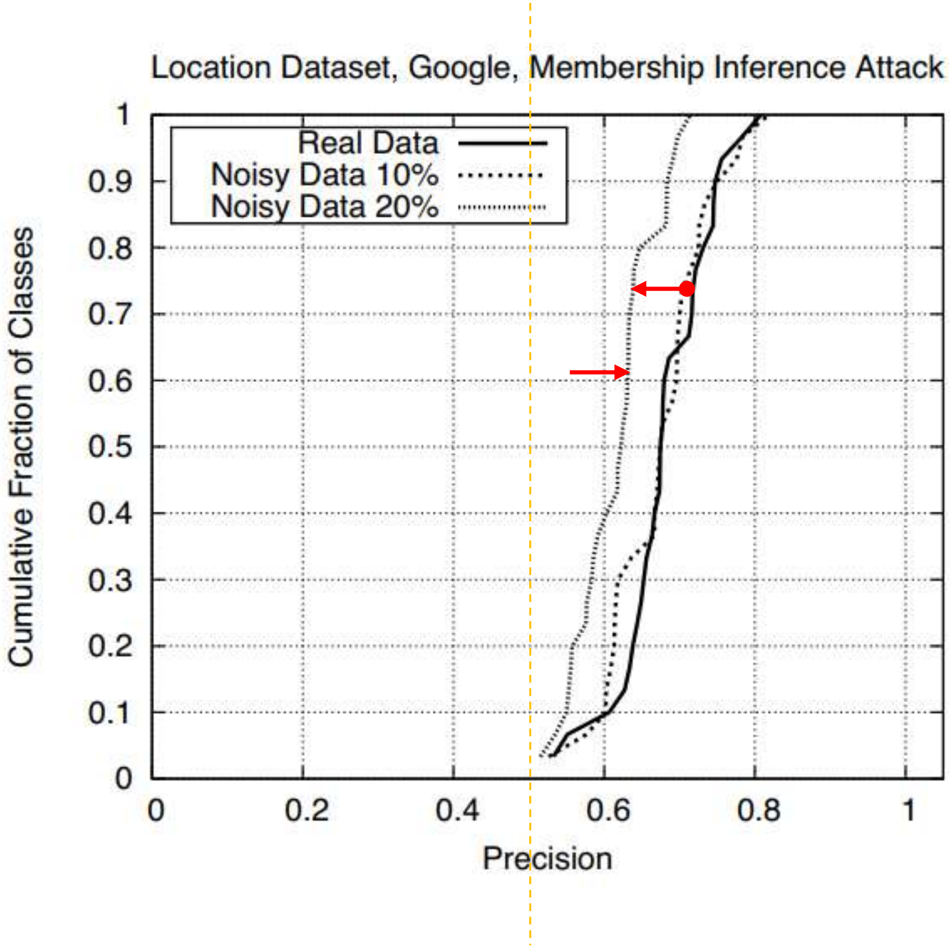


[O3] Google Prediction API leaks more compare Amazon ML or NN

[O4] When the model is less over fitted, then leaks less. e.g. NN model

[O5] Overfitting is not the ONLY reason for information leakage.
The model structure & architecture are also the reason for information leakage.
E.g. Google P-API vs Amazon ML

[R3] Performance with Noisy Data

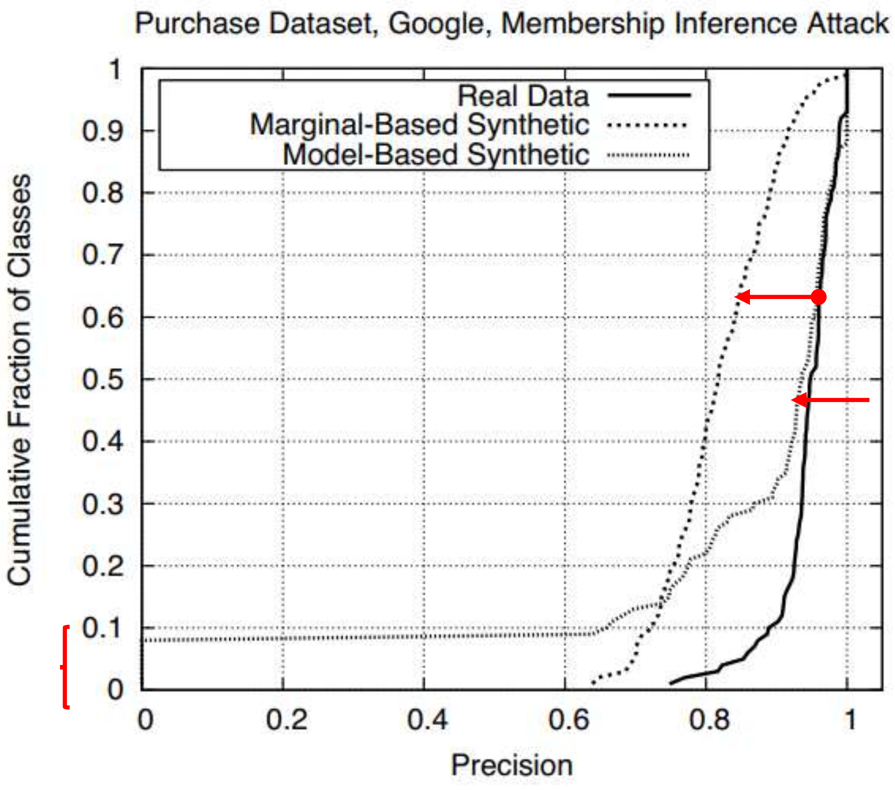


[O6] With the increase in noise in attack model's training data, it's performance drops

[O7] Even when the noise level is 20%, the attack model outperforms baseline (0.5).

*** This indicates that the proposed model is robust even when the adversary's assumption about target model's training data is not accurate**

[R4] Performance with Synthetic Data



Less than 0.1 fraction of classes performs poor
These classes contributes under 0.6% of the training data

[O8] There is a significant performance drop when marginal based synthetic data is used

[O9] Attack using model-based synthetic data performs closer to attack using real data except for classes with less training examples in target model's training data.

*** Membership inference attack is possible only with black-box access to target model without any knowledge about data population**

Performance in Six Datasets

Attack performance is low
1. [O2] K is low
2. [O4] Low Overfitting

Dataset	*Training Accuracy*	*Testing Accuracy*	*Attack Precision*
Adult	0.848	0.842	0.503
MNIST	0.984	0.928	0.517
Location	1.000	0.673	0.678
Purchase (2)	0.999	0.984	0.505
Purchase (10)	0.999	0.866	0.550
Purchase (20)	1.000	0.781	0.590
Purchase (50)	1.000	0.693	0.860
Purchase (100)	0.999	0.659	0.935
TX hospital stays	0.668	0.517	0.657

Overfit/Performance Gap

0
0.06
0.33
0.01
0.13
0.22
0.31
0.34
0.15

Attack performance is high
1. [O4] High Overfitting

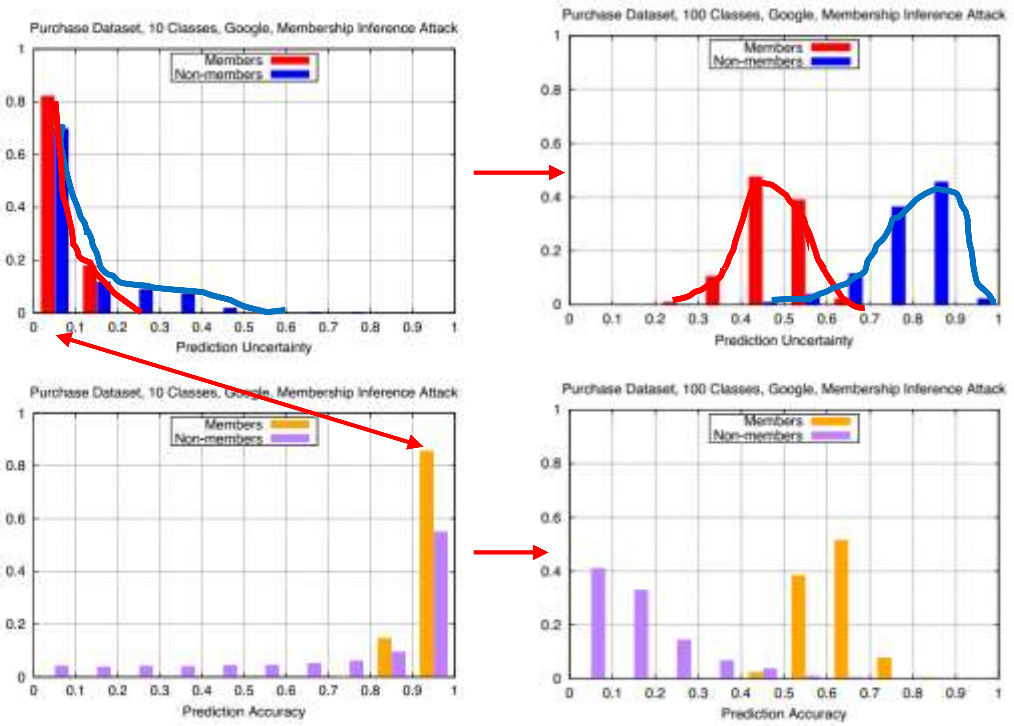
Attack performance is low even with high overfitting
[O5] Overfitting is not the ONLY reason for information leakage

[R5] Why does attack successful?

- Prediction Uncertainty vs Accuracy of Target Model

  PU = Normalized entropy of the prediction vector

Fraction of test sample



K = 10　　K = 100

[O10] When PU is low, the accuracy of target model is high

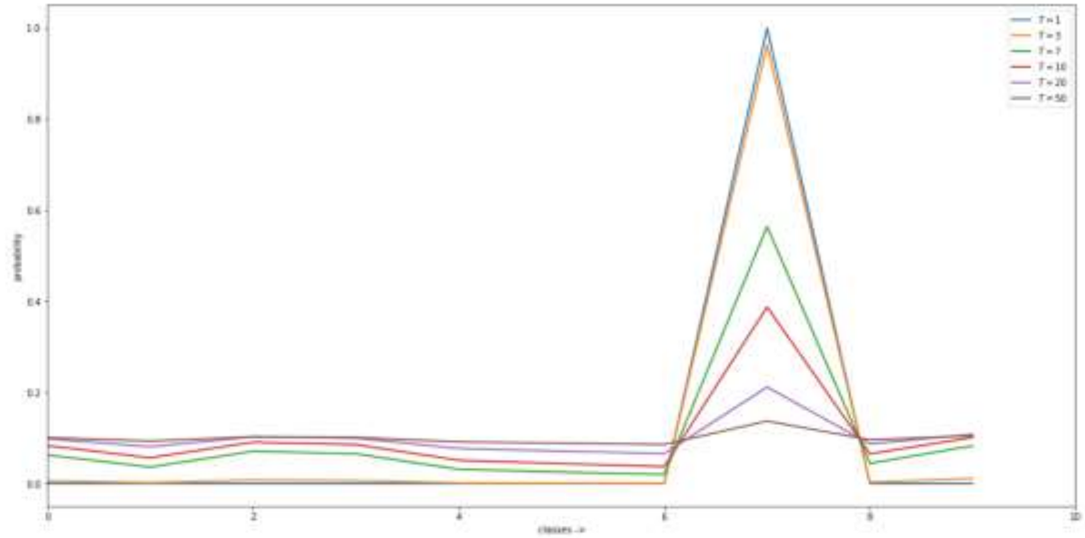
[O11] With the increase in K, PU increases & accuracy of target model drops

[O12] As K increases, PU distribution is significantly differ for member vs non-members

*** Behavioral difference in target model for a member vs non-member is utilized by the attack for successful attack**

Mitigation Strategies

1. Restrict the prediction vector to top k classes, in most restrictive scenario return only the label of top most likely class
2. Round up probabilities to d digits
3. Increase entropy of prediction vector
   - E.g. Apply a temperature variable to SoftMax layer of NN
4. Regularization – penalize for larger r



[R6] Effect of Mitigation 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

[O13] Precision drops significantly, when top 1 label ONLY is returned or with high regularization

[O14] Accuracy drops significantly, when t=20 or with high regularization

[O15] Target model's performance is even increased with required amount of regularization

[O16] High regularization may significantly reduce the target model's performance

[O17] Even with the mitigation strategies, attack model outperforms baseline (0.5)

*** Attack is robust**

Conclusion

- **Success of** the membership inference **attack** is **depended on**
  1. **Generalizability** of the target model
  2. **Diversity of** the training **data**
- **Overfitting is** one of the important reason for information leakage, but **not the ONLY reason**. **Model type & training also determines** the amount of information leakage
- **Membership inference attack can be successful with black-box access** to target model even if the adversary has no knowledge about data population

Thank You