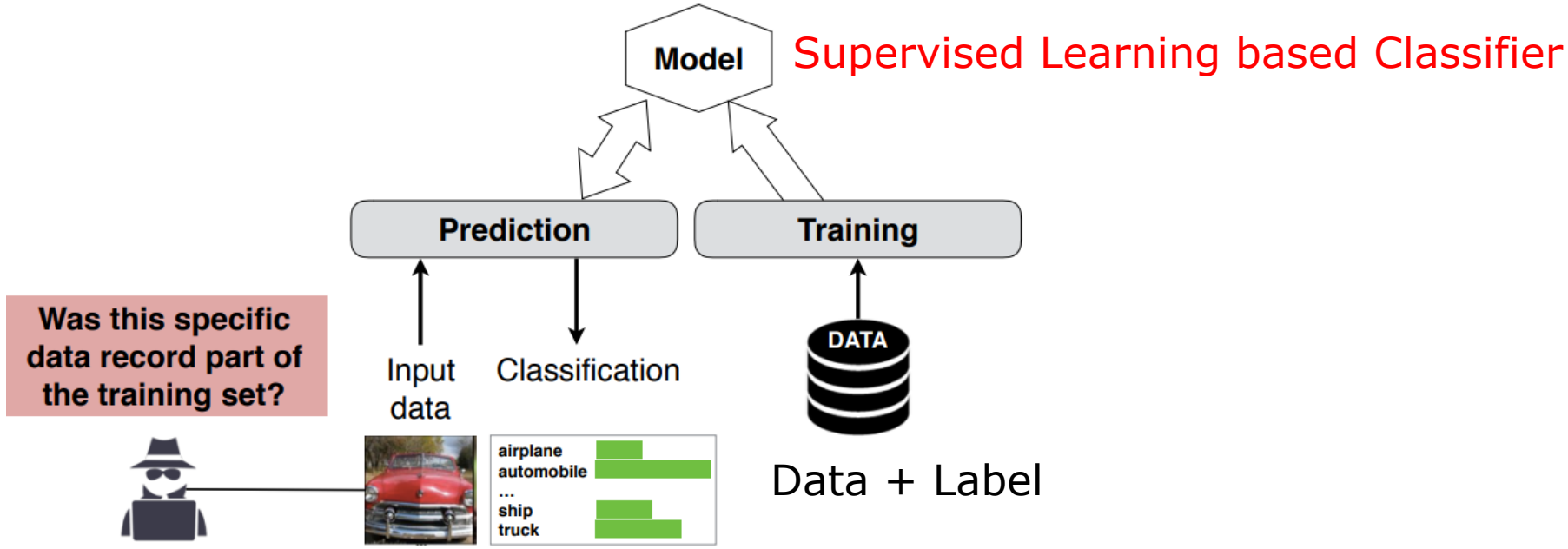
# Membership Inference Attacks against Machine Learning Models (S&P'17)

Reza Shokri, Marco Stronati,
Congzheng Song, Vitaly Shmatikov
Presenter: Kiwon Chung

# Introduction

- What is Membership Inference Attack?



# Introduction

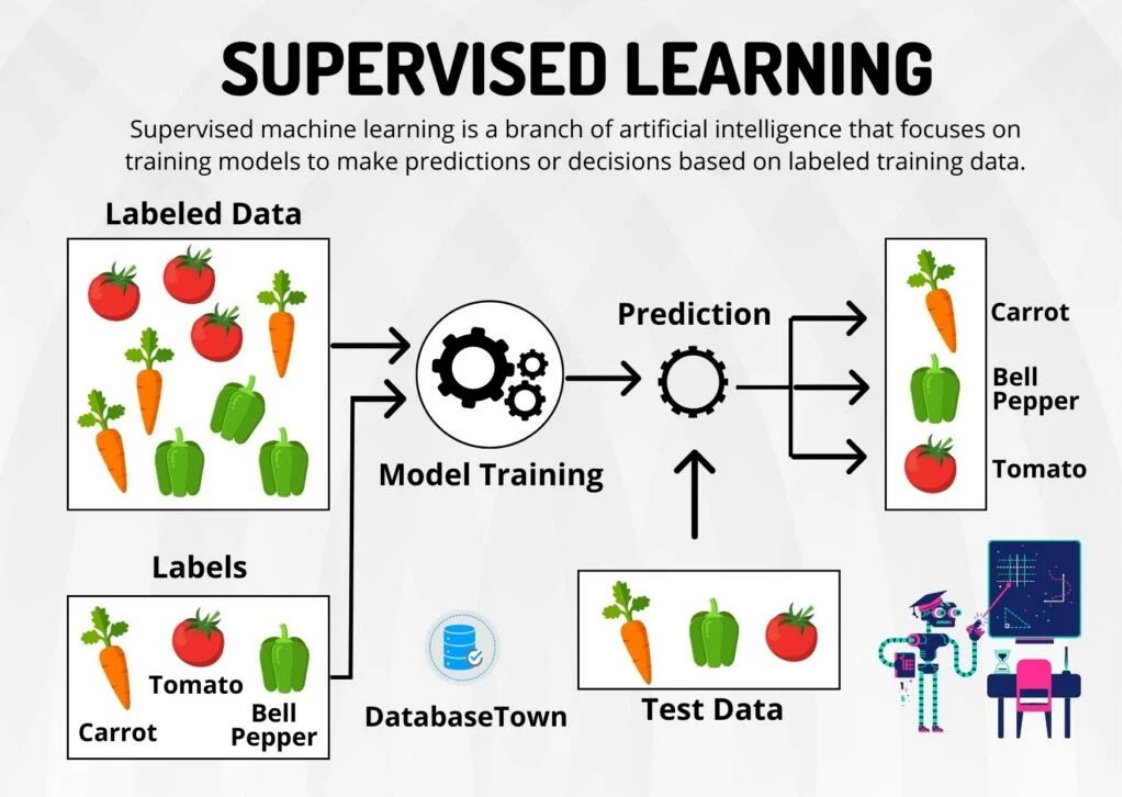
- How Membership Inference Attack can happen?
  - Service providers do not warn customers about **overfitting**
    - Google Prediction API hides all details
    - Amazon ML provides only a very limited set of pre-defined options
  - Model type & architecture

# Introduction

- Contributions of the paper
  - Propose "**Black-box**" Membership Inference Attack
  - Invented "**Shadow Training**" Technique
    - Propose algorithm for synthesizing shadow model training data
  - Conducted experiments on Real-World MLaaS
    - Amazon ML
    - Google Prediction API

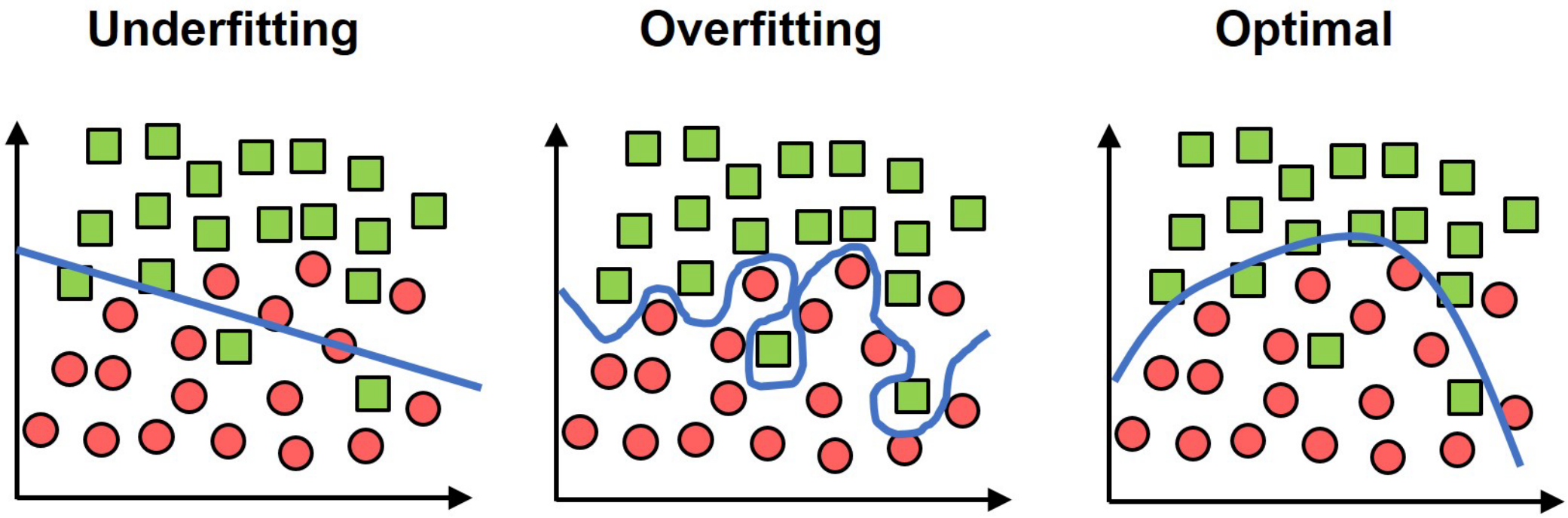
# Background

- Supervised Learning



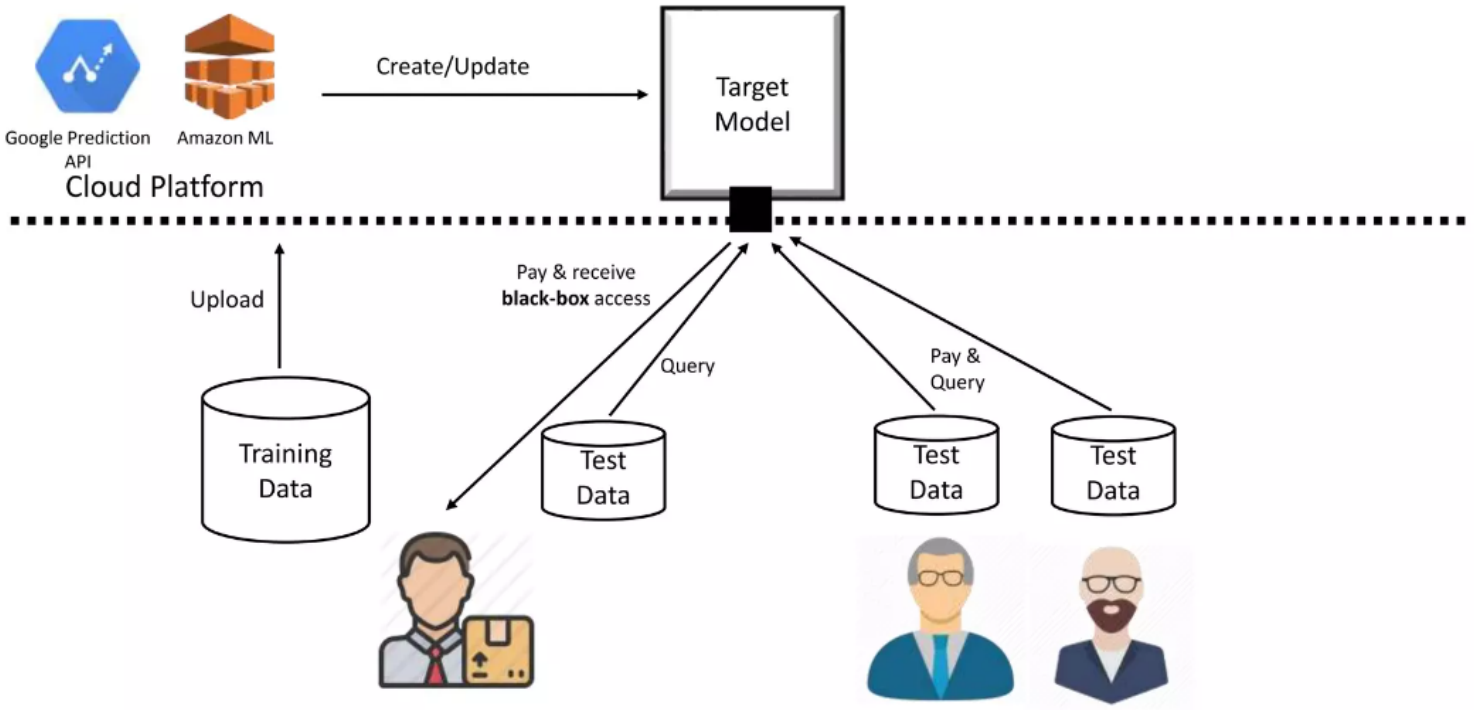
# Background

- Overfitting



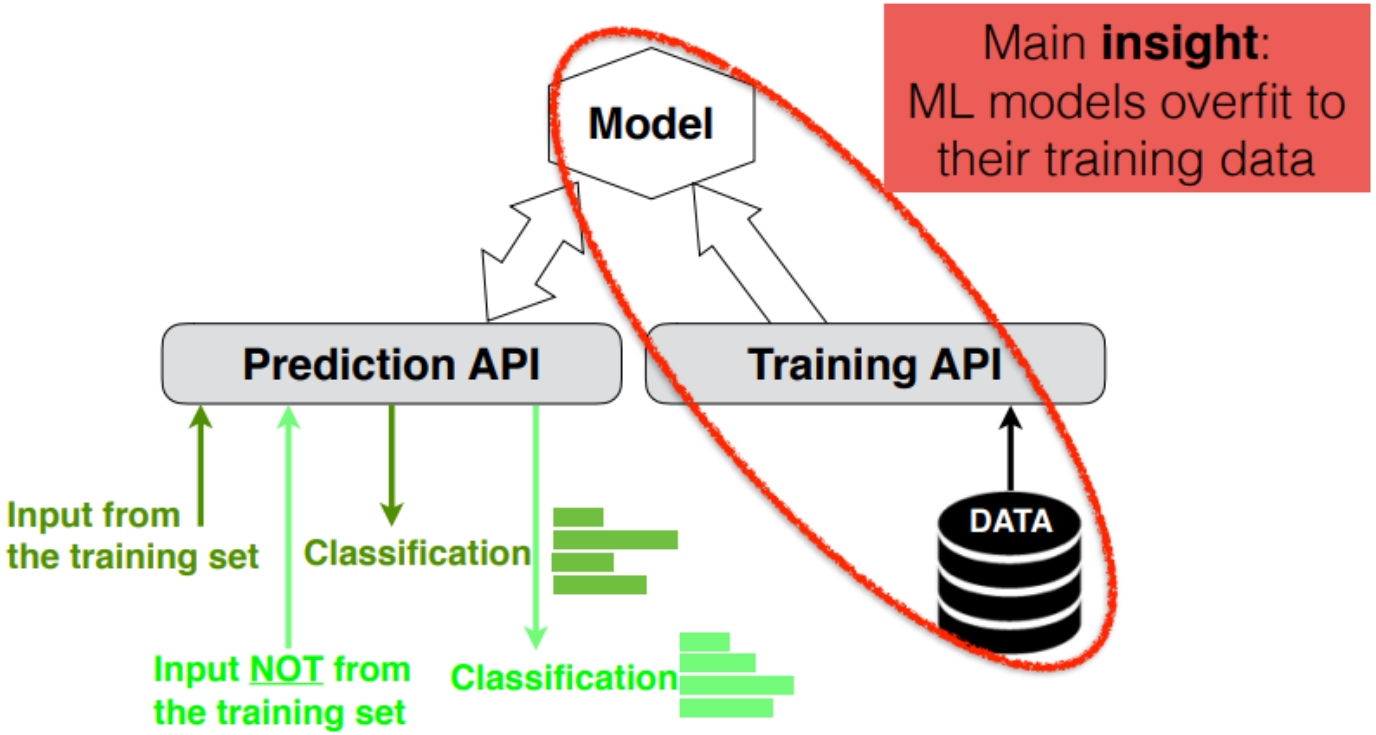
# Background

- Machine Learning as a Service (MLaaS)



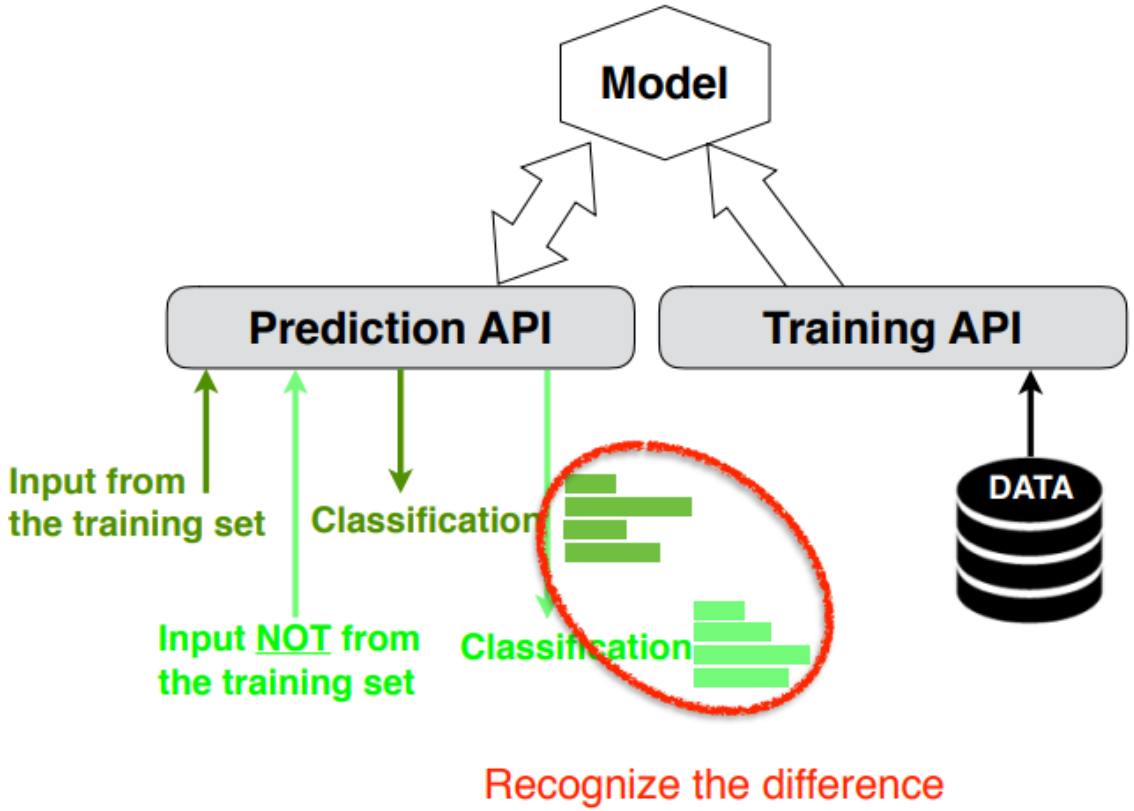
# Methodology

- Exploit Model's Predictions
  - Attacker can earn prediction vector



# Methodology

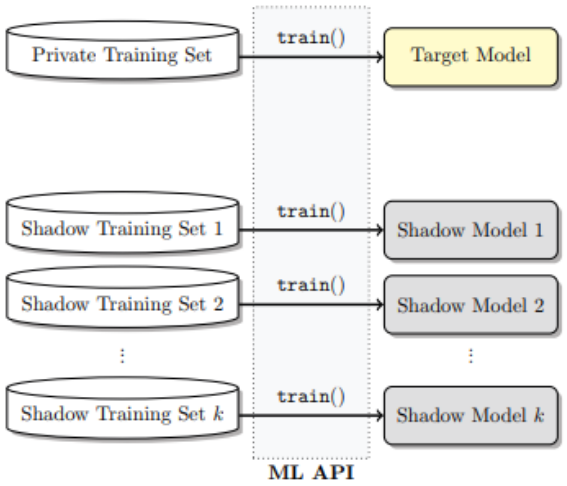
- Exploit Model's Predictions




How to make Member/Non-Member Data?

# Methodology

- Train Shadow models with Synthesized Datasets
  - Attacker knows the format of inputs & outputs of target model
  - Attacker knows the architecture of target model
    - Generate Shadow models with same architecture



# Methodology

- Strategies for Data Generation
  - Varies on Range of Knowledge of the Adversary
    - Model-based Synthesis
    - Statistics-based Synthesis
    - Noisy Real Data

# Methodology

- Model-based Synthesis
  - Attacker has neither real t bution statistics
  - Intuition: Records that are del with high confidence should be rget's training dataset

**Algorithm 1** Data synthesis using the target model

```
1:  procedure SYNTHESIZE(class : c)
2:      x ← RANDRECORD( )          ▷ initialize a record randomly
3:      y*_c ← 0
4:      j ← 0
5:      k ← k_max
6:      for iteration = 1 ··· iter_max do
7:          y ← f_target(x)                ▷ query the target model
8:          if y_c ≥ y*_c then             ▷ accept the record
9:              if y_c > conf_min and c = arg max(y) then
10:                 if rand() < y_c then                ▷ sample
11:                     return x                        ▷ synthetic data
12:                 end if
13:             end if
14:             x* ← x
15:             y*_c ← y_c
16:             j ← 0
17:         else
18:             j ← j + 1
19:             if j > rej_max then        ▷ many consecutive rejects
20:                 k ← max(k_min, ⌈k/2⌉)
21:                 j ← 0
22:             end if
23:         end if
24:         x ← RANDRECORD(x*, k)          ▷ randomize k features
25:     end for
26:     return ⊥                           ▷ failed to synthesize
27: end procedure
```

# Methodology

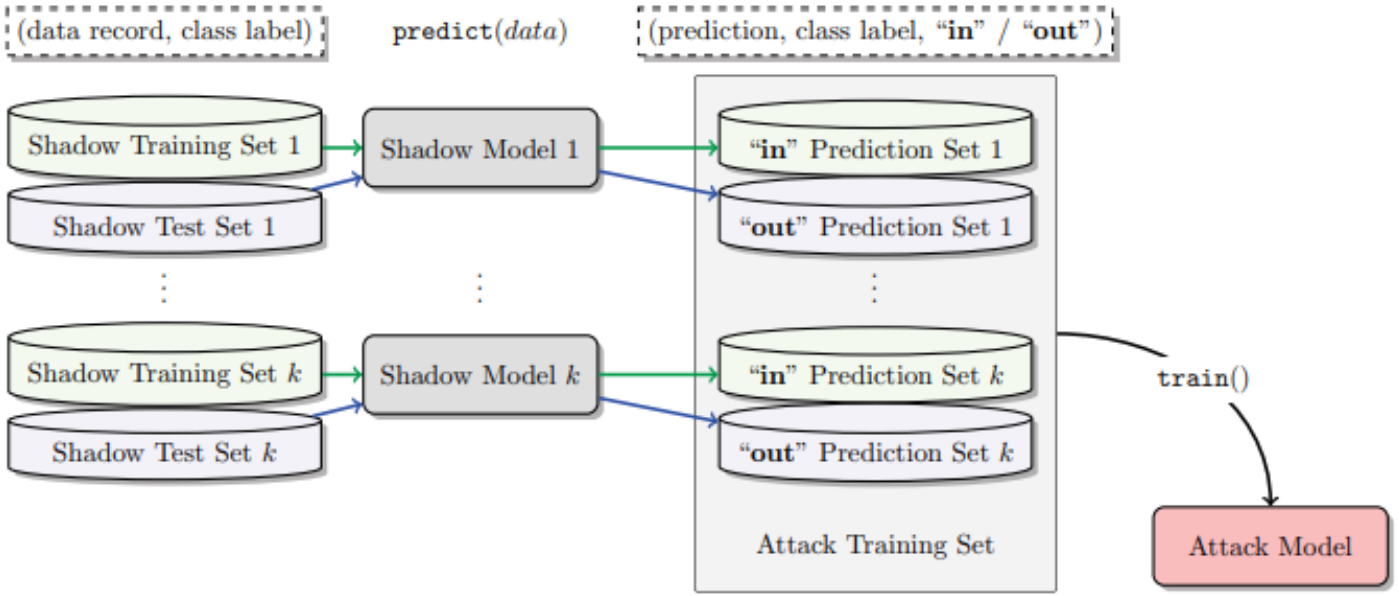
- Statistics-based Synthesis
  - Attacker may have prior knowledge of the marginal distributions of different features
  - Generate synthetic training records for the shadow models by independently sampling the value of each feature from its own marginal distribution

# Methodology

- Noisy Real data
  - Attacker may have access to some data that is similar to the target model's training data and can be considered as a "noisy" version
  - Authors simulated by flipping the (binary) values of 10% or 20% randomly selected features in 'location' dataset

# Methodology

- Training Attack Model
  - At
  - Qu
  - Qu
  - Co
  - Fo




wn training

model and earn

del and earn

dow models to

l which predicts

'**in**' or '**out**' with provided dataset (triplets)

# Evaluation

- Datasets
  - CIFAR (CIFAR-10/CIFAR-100)
    - 32×32 color images
    - CIFAR-10: 50k/10k Train/Test instances for each 10 classes
    - CIFAR-100: 500/100 Train/Test instances for each 100 classes
  - Purchases
    - Based on Kaggle's "acquire valued shoppers" challenge dataset
    - 600 binary features, 197,324 instances
    - {2,10,20,50,100} classes
    - 10k Train instances, Rest instance to Test instances & Shadow training set

# Evaluation

- Datasets
  - Locations
    - 11,592 users and 119,744 locations, 1,136,481 check-ins
    - 446 binary features
    - 30 classes
    - 1.6k Train instances, Rest instances to Test instances & Shadow training set
  - Texas Hospital Stays
    - 6,170 binary features, 67,330 instances
    - 100 classes
    - 10k Train instances, Rest instances to Test instances & Shadow training set

# Evaluation

- Datasets
  - MNIST
    - 32×32 Black & White images
    - 10 classes
    - 10k Train instances, 60k instances to Test instances & Shadow training set
  - UCI Adult
    - 14 attributes, 48,842 instances
    - Binary class
    - 10k Train instances, Rest instances to Test instances & Shadow training set

# Evaluation

- Target models
  - Offline Neural Networks
  - Google Prediction API
    - No configuration can be changed by user
  - Amazon ML
    - Can control few configurations
      - # of epochs & Regulation amount

# Evaluation

- Dataset & Target Model Summary

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

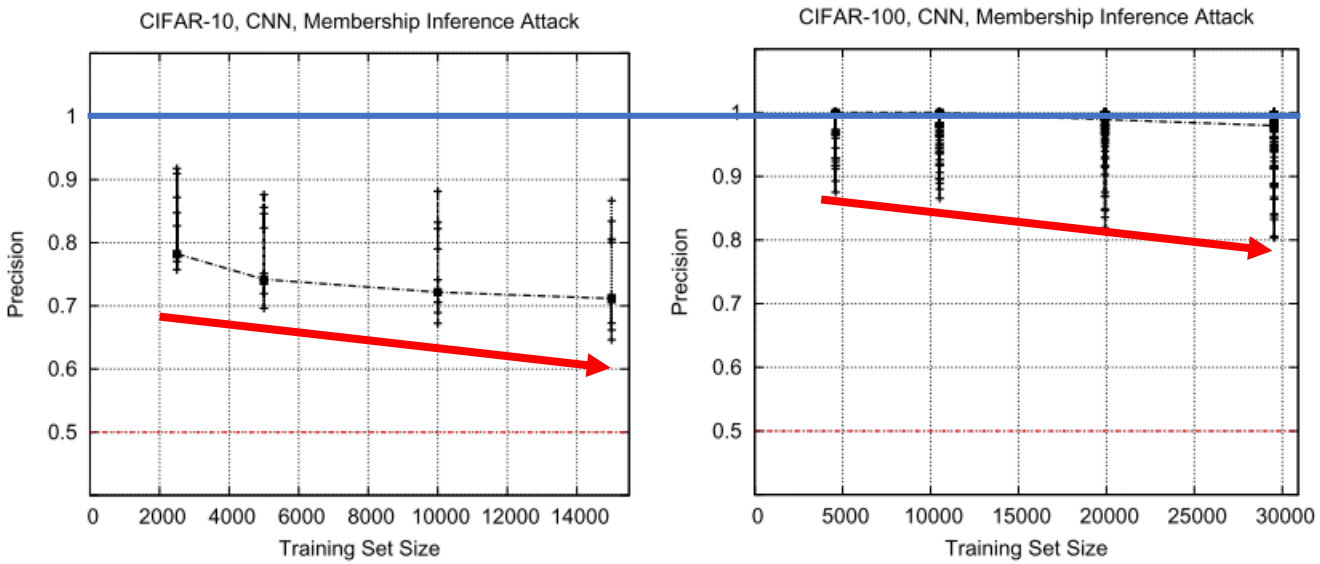
Google P-API : no control of the training
Amazon ML : # of epochs & regularization amount are changed
Two set of configurations of Amazon ML: (10, 1e-6) & (100, 1e-4)

# Analysis & Results

- Attack settings
  - Test Set: 50% members & 50% non-members of target model
    - Baseline Accuracy is 0.5
  - Use Precision & Recall for metrics

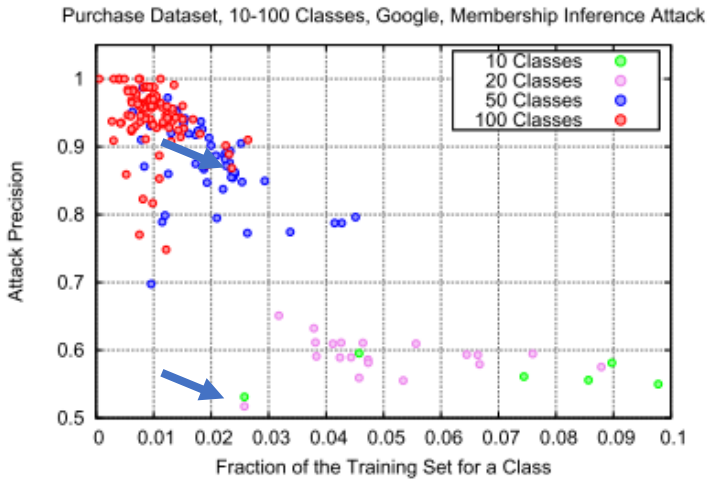
# Analysis & Results

- [R1] Label & Training set size



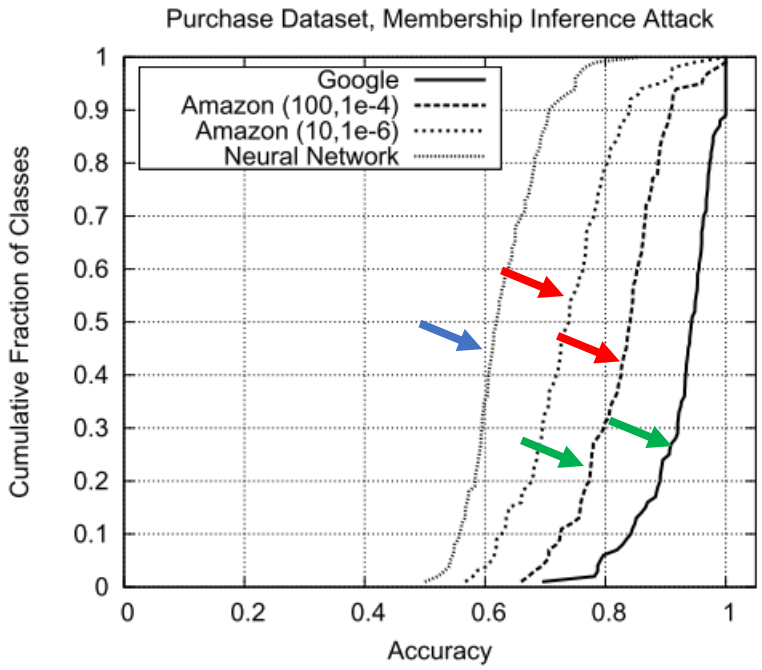

1) As Label Size ↑, Precision ↑

2) As Training set Size ↑, Precision ↓



# Analysis & Results

- [R2] Overfitting & Model Types




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

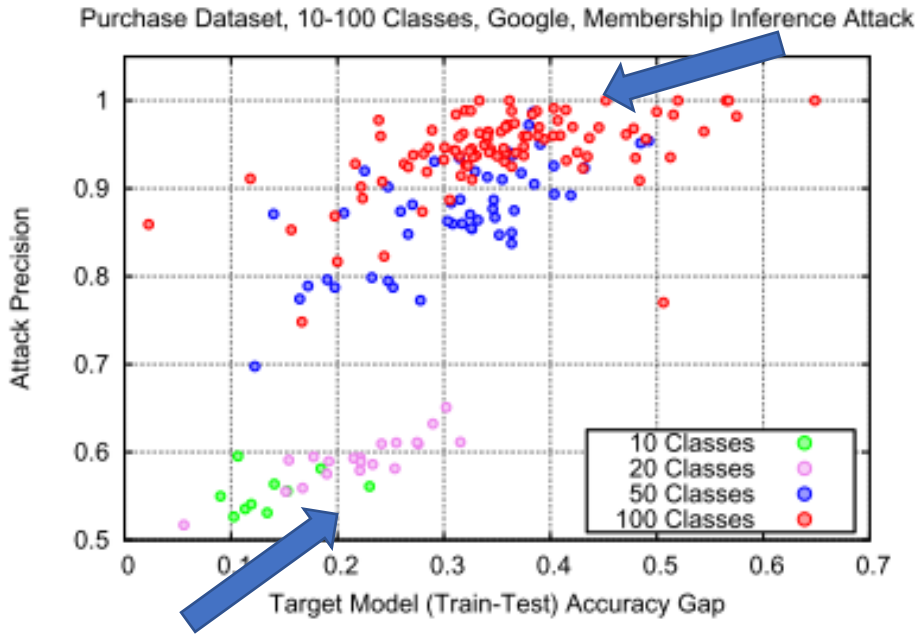
1) In Amazon ML service, More overfitted model (100) leaks more

2) The least overfitted model (NN) leaks least

3) Comparing with Google & Amazon-100, Overfitting is not the **ONLY** reason

# Analysis & Results

- [R2] Overfitting & Model Types



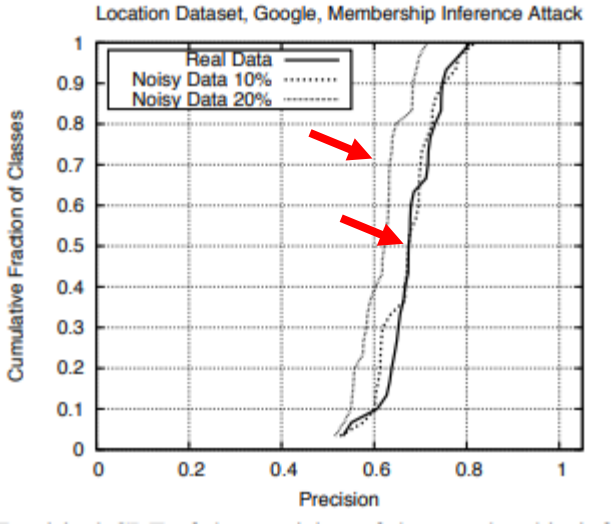

| *Dataset* | *Training Accuracy* | *Testing Accuracy* | *Attack Precision* |
|---|---|---|---|
| Adult | 0.848 | 0.842 | 0.503 |
| MNIST | 0.984 | 0.928 | 0.517 |
| Location | 1.000 | 0.673 | 0.678 |
| Purchase (2) | 0.999 | 0.984 | 0.565 |
| Purchase (10) | 0.999 | 0.866 | 0.550 |
| Purchase (20) | 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 |

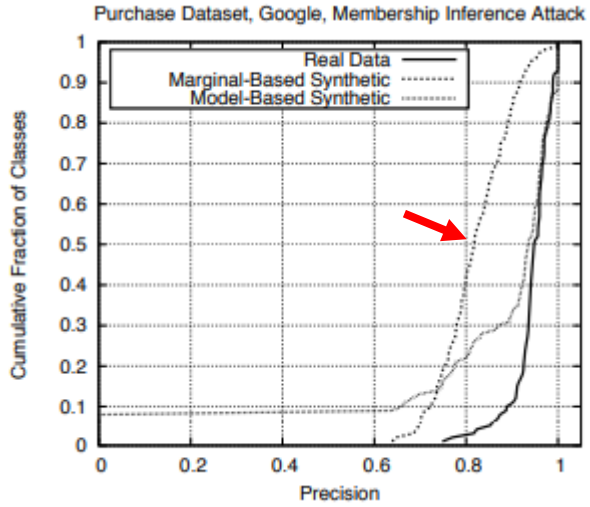
TABLE II: Accuracy of the Google-trained models and the corresponding attack precision.

Overfitting is not the **ONLY** reason

# Analysis & Results

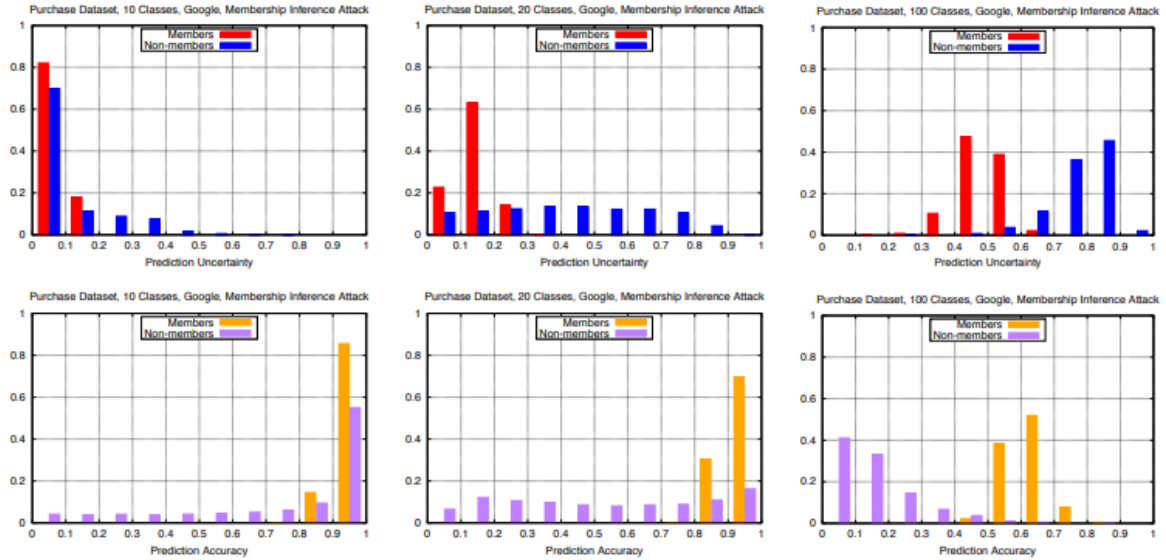
- [R3] Synthesis strategy
  - Noisy Data (Location)
  - Marginal-Based & Model-Based (Purchase)
    - Model-Based Strategy is better than Marginal-Based



# Analysis & Results

- [A1] Why Attack works?
  - As Class label size increases, Prediction Uncertainty distribution can be easily distinguished between Members & Non-Members



# Mitigation Strategies

- Restrict Prediction Vector to top-k classes
- Round up probabilities to d digits
- Increase entropy of prediction vector
  - Apply a temperature variable to Softmax Layer
- Regularization

# Effects of Mitigation Strategies

- Using Top-K label strategy can reduce **precision** effectively
- Using Temperature t=20 can reduce **recall**
- Regularization strategy can reduce **precision** & **recall** both, but it significantly reduce test accuracy

| **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 |

# Summary

- Success of Membership Inference Attack is depend on
  - Prediction Uncertainty (Generalization ability of Target Model)
  - Large label size with small training size for each label (Diversity of training dataset)
- Overfitting is important, but not the only reason

# Conclusions

- Strengths of the paper
  - Proposed Black-box Attack
  - Shows Attack on Real World MLaaS
  - They used various datasets with image & tabular domain

# Conclusion

- Limitations of the paper
  - Authors didn’t explain how they select these shadow model numbers for each dataset
  - Authors didn’t examine deeply that what model architecture is related with membership inference attack (Lack of Ablation Study)
    - Can varying layer number, activation function, batch-norm …
  - Strong assumptions for attacker
    - Information of target model
    - Synthesis strategy (Marginal & Noisy data)

# Conclusion

- Some research questions I had
  - How about label only model?
  - Do we really need shadow model?
  - Are there any better data synthesis method like GAN?
  - How do we know what kind of architecture or models are prone to Membership Inference Attack?
  - Can we boost distribution difference between Mem/Non-Mem?
  - Are there any optimization based defense method?
  - Can we also apply the attack on other domain?

# Related Works

- Backes et al. (CCS'16) Membership Privacy in MicroRNA-based Studies
  - Membership Inference Attack on Summary Statistics using Likelihood ratio test
- Yeom et al. (CSF'18) Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting
  - Infers membership by comparing the ground truth against the predicted label
- Salem et al. (NDSS'19) ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models
  - Infers membership by applying a simple threshold for predicted label's confidence

# Related Works

- Jia et al. (CCS'19) MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples
  - By utilizing adversarial examples, randomly select training data points to add noise into confidence vector
- Hui et al. (NDSS'21) Practical Blind Membership Inference Attack via Differential Comparisons
  - Generates set of Non-Members and exchange with target datasets 1-on-1
  - Compare the distance between Before & After 1-on-1 exchange to determine Mem/Non-Mem
- Choquette-Choo et al. (ICML'21) Label-Only Membership Inference Attacks
  - Predict that data points that exhibit high robustness are training data points
  - Using 'HopSkipJump' to calculate decision boundary distance as a measure of robustness

# Related Works

- Carlini et al. (S&P'22) Membership Inference Attacks From First Principles
  - Conduct Likelihood Ratio Attack with new logit scaling function
  - Emphasize False-Positive rate as an important metric
  - Measures hardness per class & example
- Tramèr et al. (CCS'22) Truth Serum: Poisoning Machine Learning Models to Reveal Their Secrets
  - Using Poisoning attack (<0.1%) to separate confidence distribution of Mem/Non-Mem
  - Boost attacks by 1 to 2 orders of magnitude

# Good Questions

- Is this membership information attack possible in the Large Language Model (LLM), one of the generative AIs? (김광민)

# Best Questions

- Researchers argue that machine learning service providers like Google or Amazon should consider and warn of the risk of information leaks. However, is there any concrete way to quantify the risk of this kind of problem? (허현)

# Best Questions

- What role could regulations (GDPR, etc.) or ML standards have in guiding the responsible deployment of machine learning models, especially those handling sensitive data? (cf. health-care datasets, etc.) (Valentin Guittard)

# Best Questions

- Could an ensemble method that combines predictions based on multiple models to produce the final result be one way to prevent this attack? (박승민)