Membership Inference Attacks Against Machine Learning Models

Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov

Cornell Tech, INRIA

Presentor: Joon-Gyum kim

2019/10/08

Paper Summary

Problem statement

- **Key statement:** Given a machine learning model and a record, determine whether this record was used as part of the model's training dataset or not using membership inference attack.

- **Conditions**
  - Adversary can access to model is limited to black-box queries (machine learning as a service)
  - No knowledge about the model's parameters
  - No access to internal computation of the model
  - No knowledge about the underlying distribution of data

Key Contributions

- Suggest shadow training technique that lets us train the attack model on proxy target for which we do know the training dataset
- Suggest three method to generate training data

  1) Uses black-box access to the target model to synthesize this data

  2) Uses statistics about the population from target's training dataset

  3) Assumes adversary has access to a potentially noisy version of the target's training datasee

Results

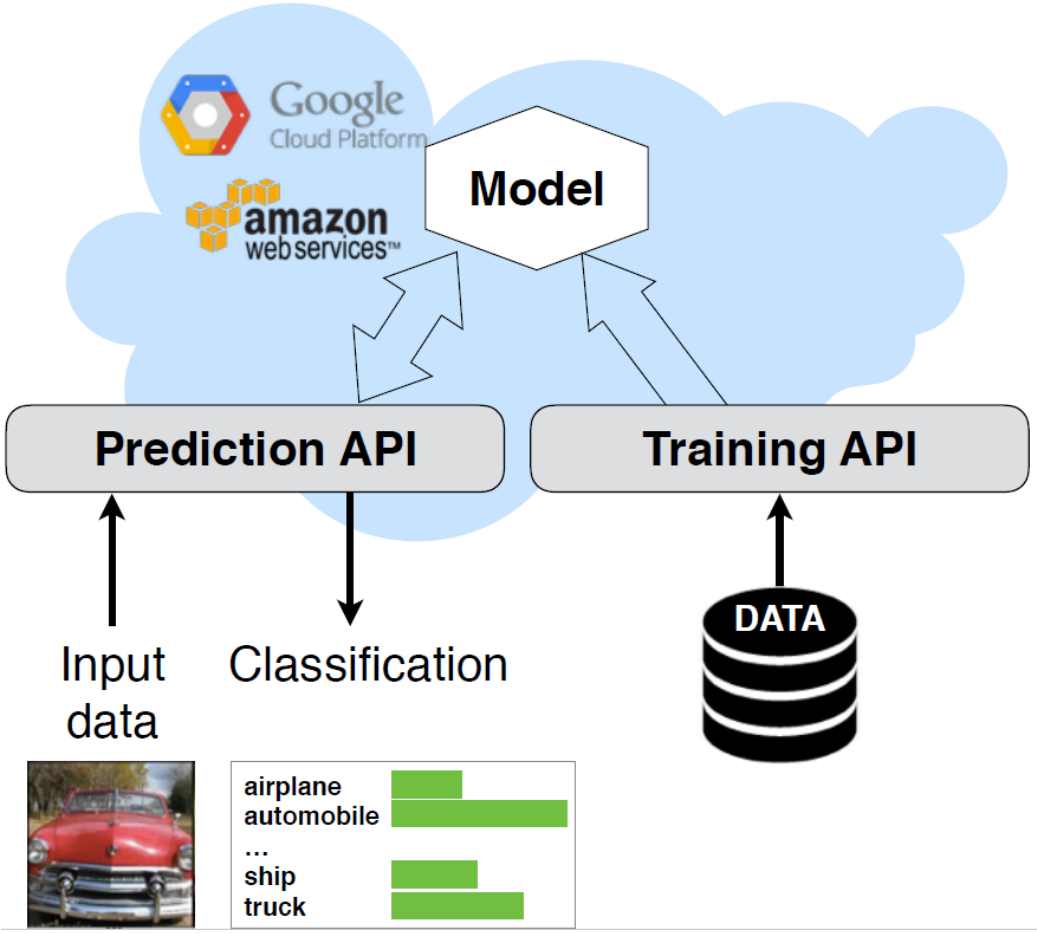
- Models from machine-learning-as-a-service platforms **leak** information about their training data
- For multi-class classification models membership inference achieves **94% (Google) and 74% (Amazon)**
- Texas hospital discharge dataset (over 70%) presents a **risk to privacy of health-care data** of individuals

Paper

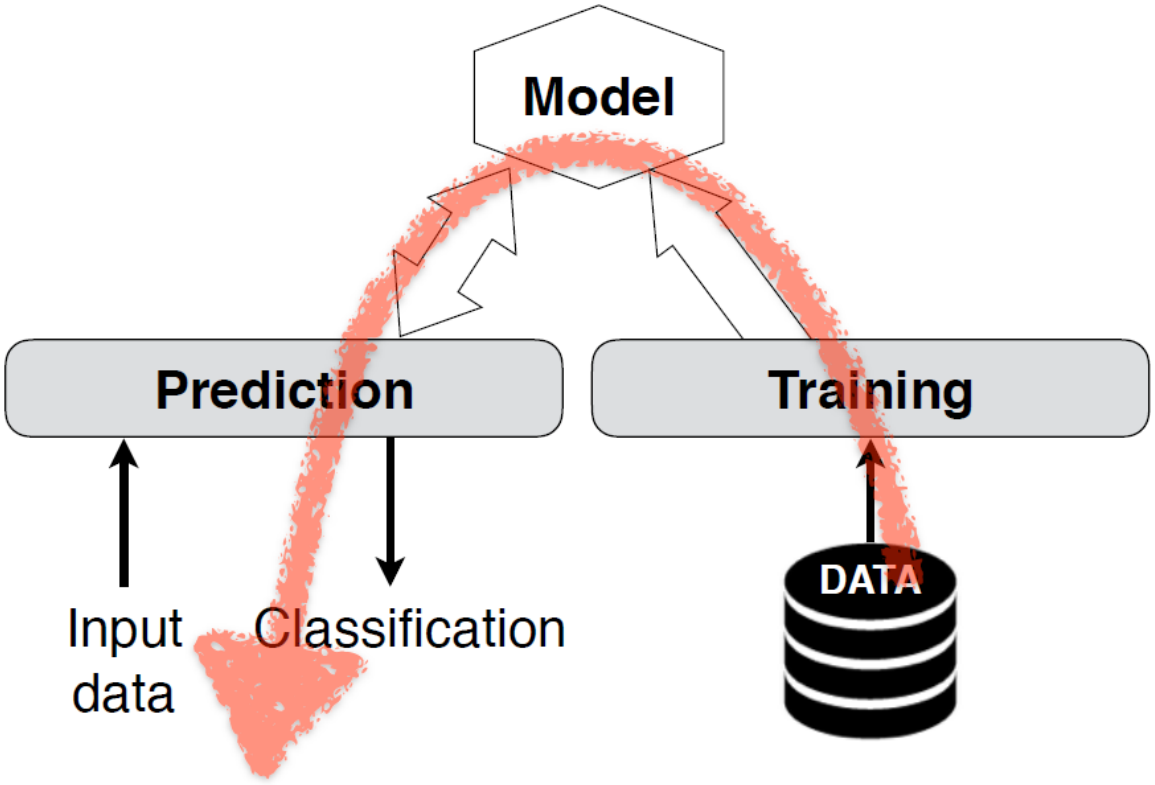
Machine-Learning-as-a-service

Black box of model

- No or little control over parameter
- User only provide training data

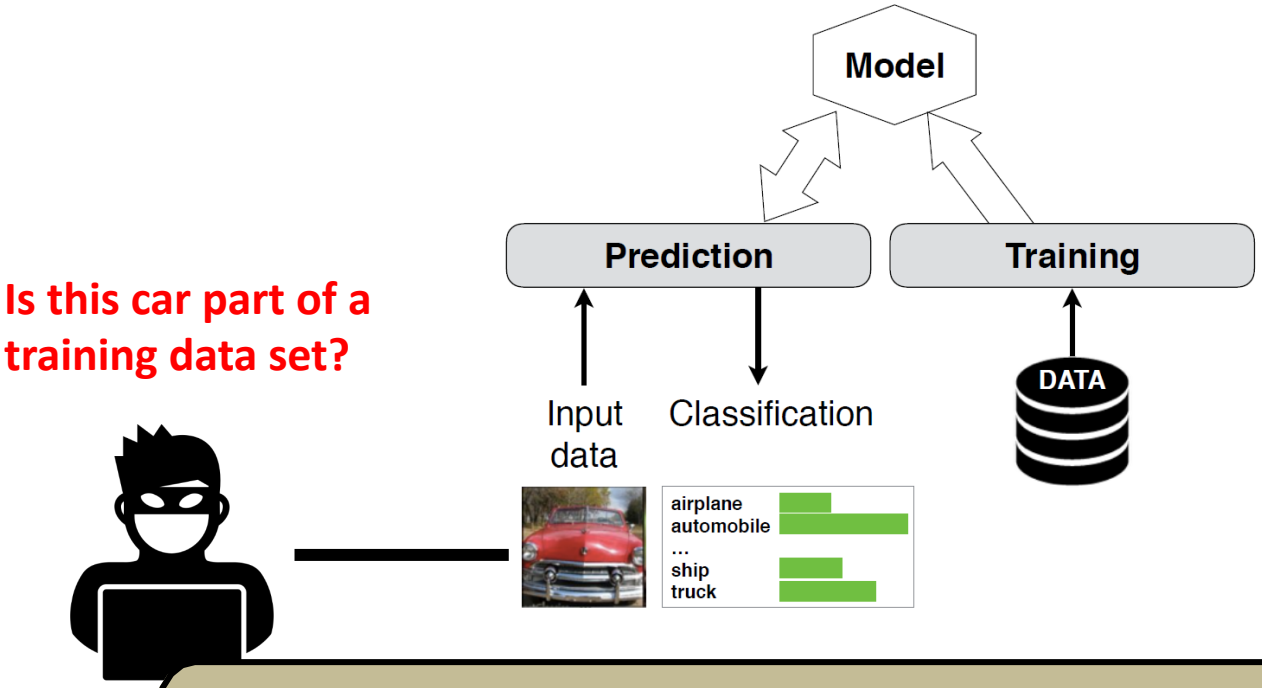


Machine Learning Privacy



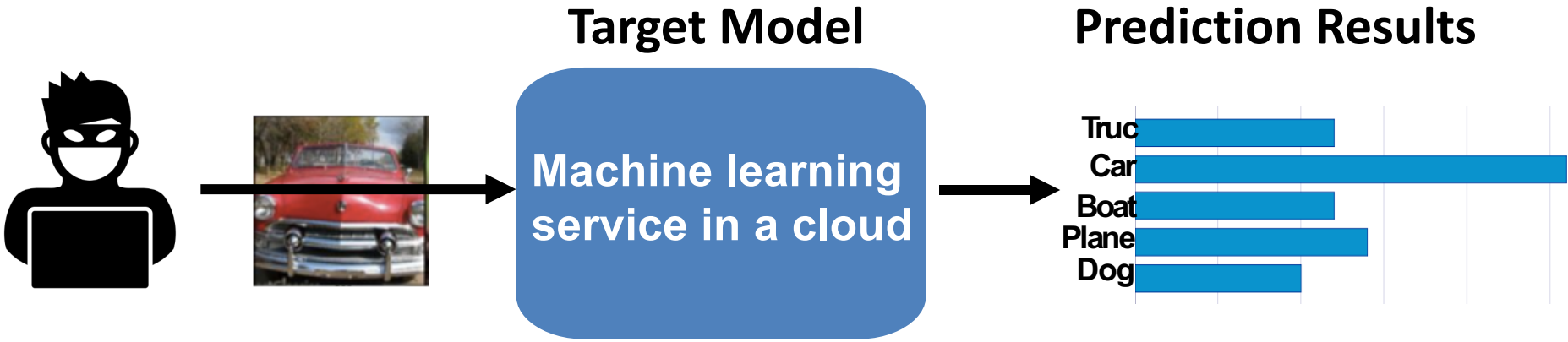
Do model's predictions leak information about training data?

Membership Inference Attack



Was this specific data record part of the training set?
=> knowing that a certain patient's clinical record was used to train a model associated with a disease can reveal that the patient has this disease

Problem Statement



Black box setting about prediction model (target model)

1. Attacker queries the target model with a data record and obtains the models prediction vector
2. No knowledge about the model's parameters
3. No access to internal computation of the model
4. No knowledge about the underlying distribution of data

Assumption about the attacker

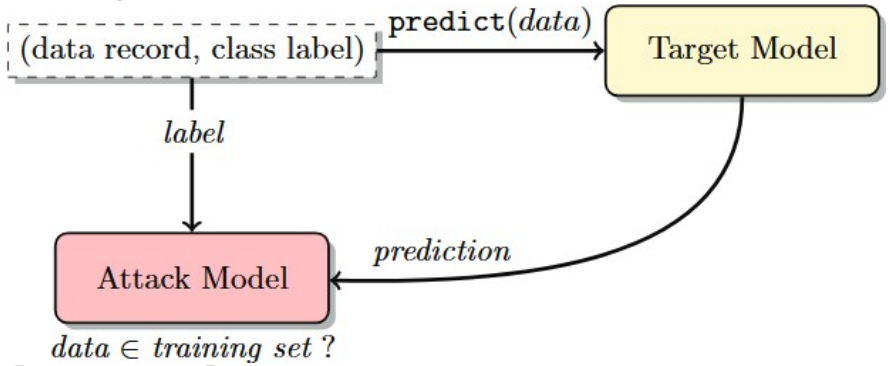
- **Assumptions about the Attacker:**
  - Query access to the model -> obtain prediction vector on any data record
  - Knowledge about input & output format of the model (including number and range of values)
  - attacker either knows (1) the type, architecture, and training algorithm of the target model, or (2) has black-box access to a machine learning oracle, that was used to train the model
  - No knowledge about model structure or meta-parameters
  - Background knowledge about the data population
  - Background knowledge about general data population statistics (Marginal distribution of feature values)

Assumption about the attacker

- **Is it a valid assumption in usual? (red assumption)**
  - Knowledge about input & output format of the model (including number and range of values)
  - attacker either knows **(1) the type, architecture, and training algorithm of the target model**, or (2) has black-box access to a machine learning oracle, that was used to train the model
  - No knowledge about model structure or meta-parameters
  - **Background knowledge about the data population**
  - **Background knowledge about general data population statistics (Marginal distribution of feature values)**

Membership Inference Overview

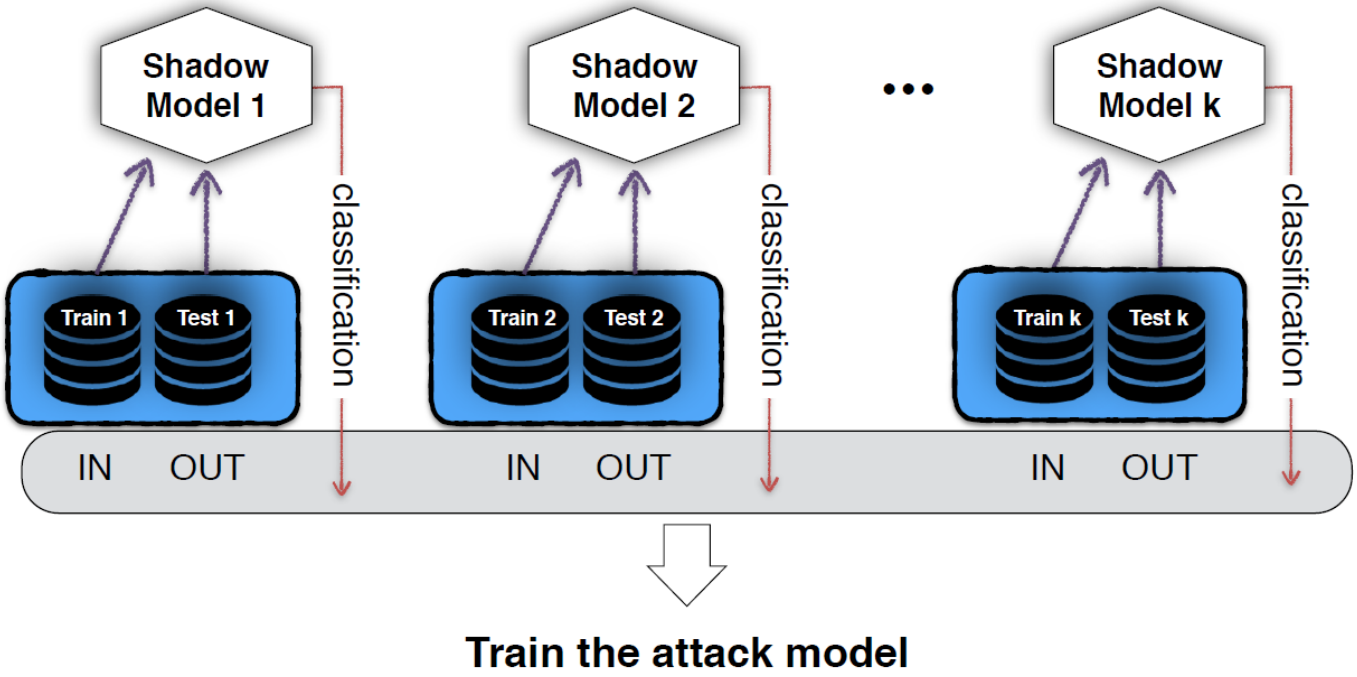
Membership inference attack in the black-box setting



- **End-to-End attack process:**
  1. Attacker queries the target model with a data record and obtains the models prediction vector
  2. Feed the prediction vector, data record and class label
  3. Attack model infers whether the record was in or out of the training dataset of the target model

The main challenge is how to train the attack model?

Shadow Models



- One shadow model for each class
- The attack will perform better if the training datasets happen to overlap
- Must be trained in a similar way to the target model

Generating Training Data for Shadow Models

- One shadow model for each class
- The attack will perform better if the training datasets happen to overlap
- Must be trained in a similar way to the target model

Generating Training Data for Shadow Models

Method 1: Model-based synthesis

- Attacker does not have real training data nor any statistics about its distribution
- Generate synthetic training querying the target model

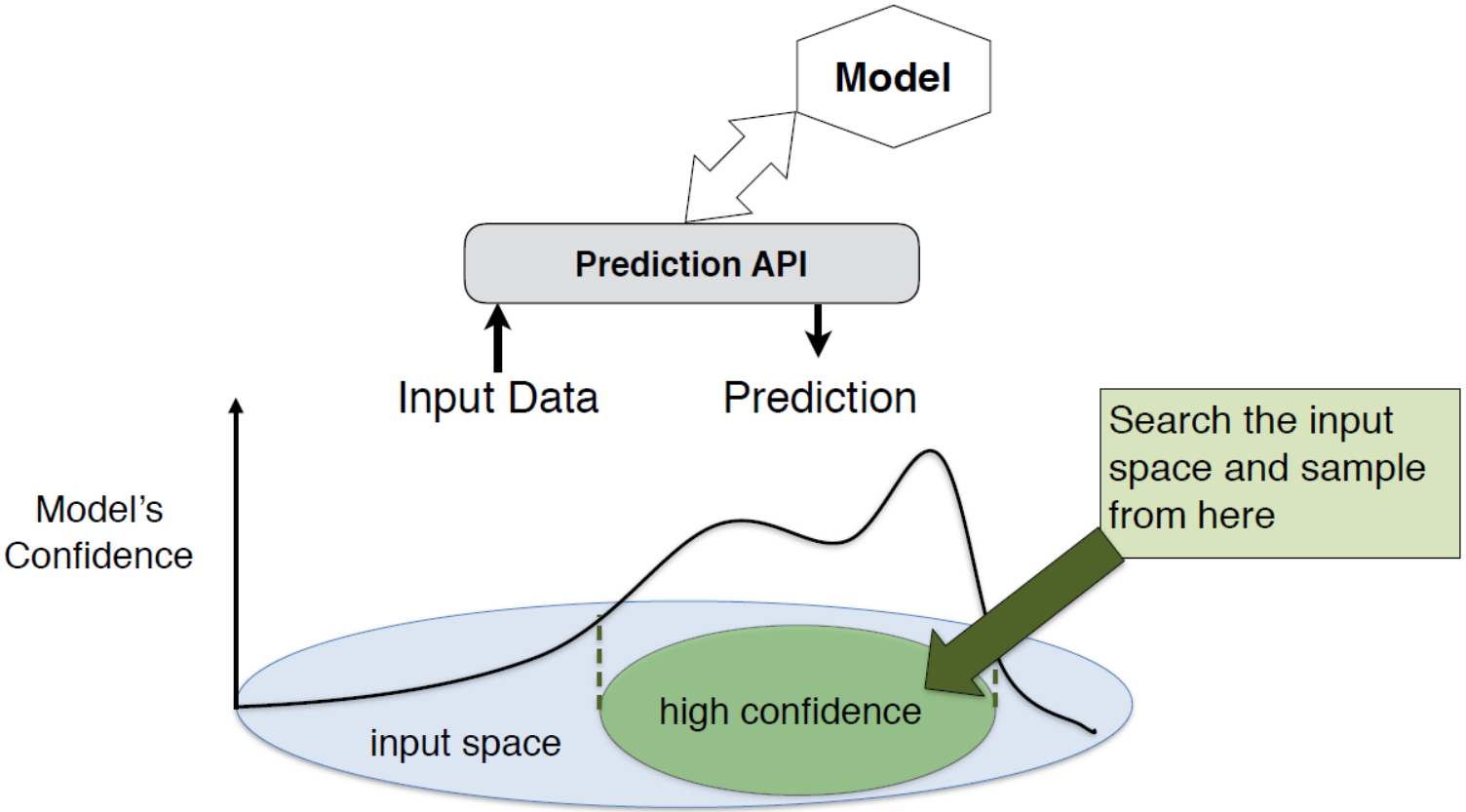
Intuition

- High confidence (synthetic) samples target model should be statistically similar to the targets training dataset

Procedure

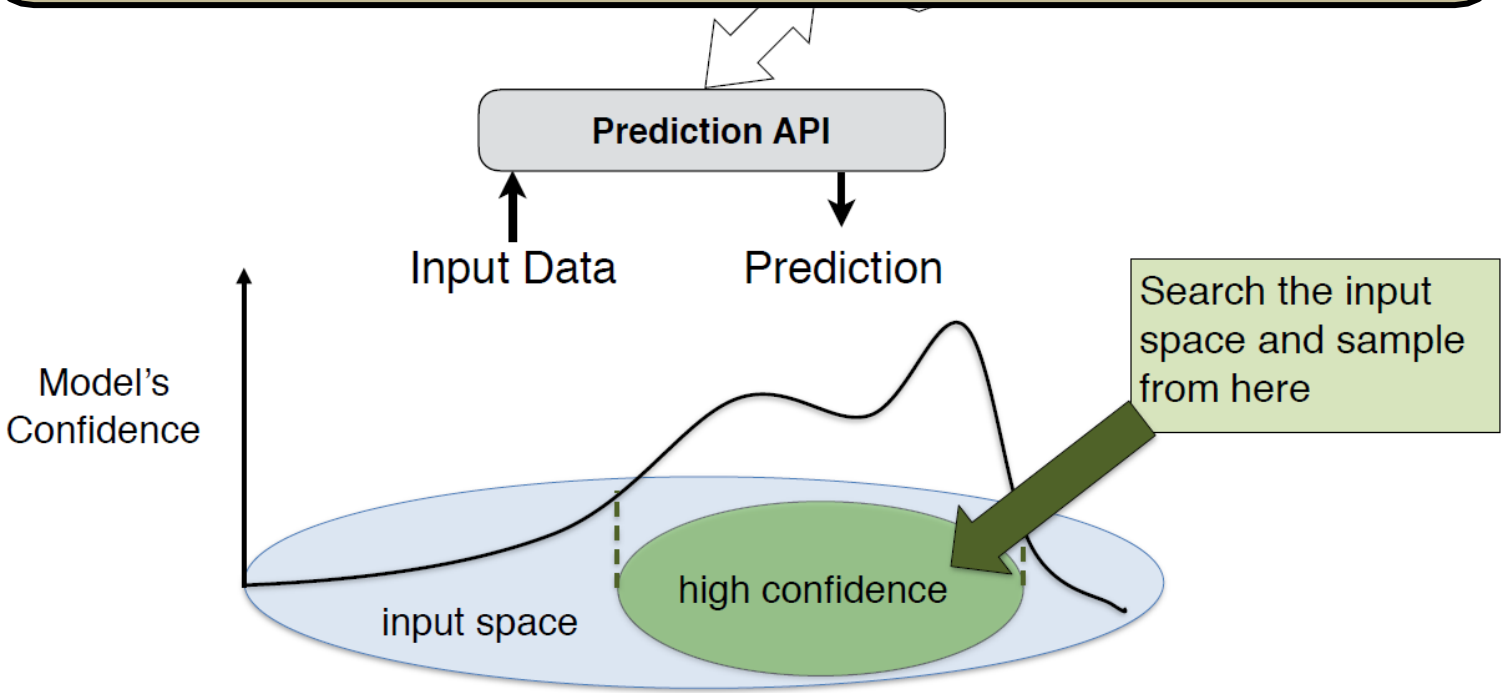
1. Search the input space and find inputs that are classified with high confidence (Hill Climbing Algorithm)
2. Sample synthetic data

Generating Training Data for Shadow Models



Generating Training Data for Shadow Models

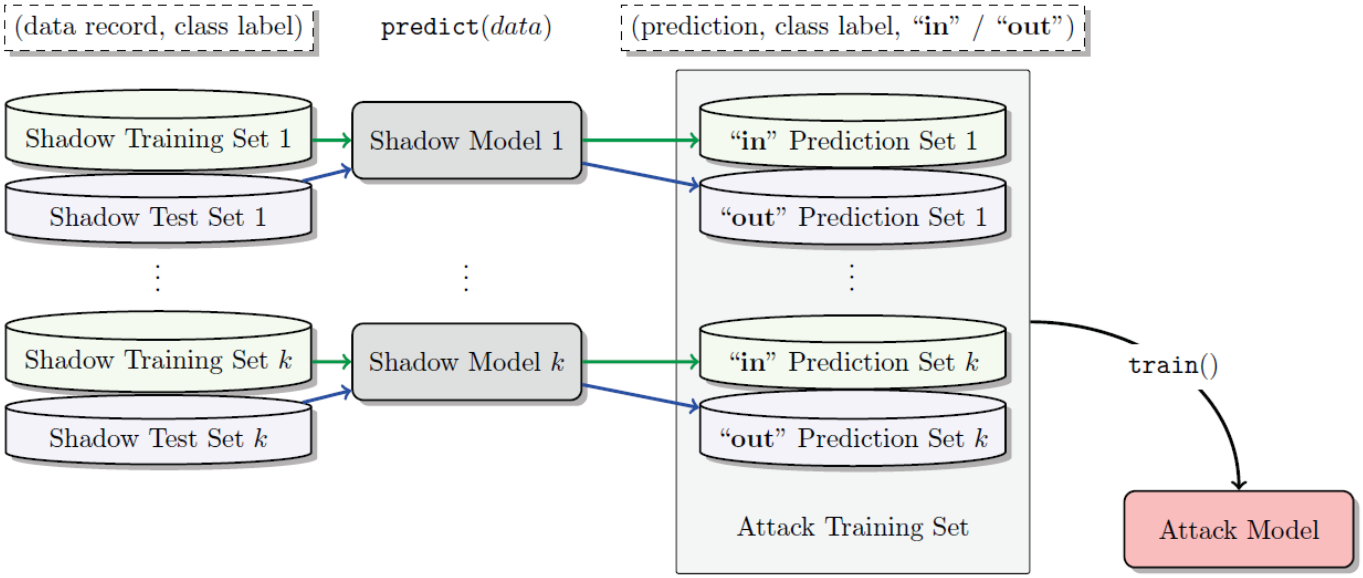
How much will this cost?



Generating training data for shadow models

- **Method 2: Statistics-based synthesis**
  - The attacker may have prior knowledge of the marginal distribution of different features
  - In experiments: Generate synthetic training records for the shadow models by independently sampling the value of each feature from its own marginal distribution
- **Method 3: 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 thereof
  - In experiments: Simulate this by flipping the (binary) values of 10% or 20% randomly selected features, before training the shadow models

Training the Attack Model



- Query each shadow model with its own training dataset and with a disjoint test set of the same size
- Generate attack training set by shadow model 1 ~ k

Evaluation

Datasets (6 Types)

- **Type1: CIFAR**
  - Image based benchmark dataset for recognition
  - 32 x 32 color image in 100 class, 6,000 images per class
  - Image classification
- **Type2: Purchases**
  - Shopping histories for several thousand individuals
  - Authors derived a simplified purchase dataset, where each record consists of 600 binary features. Each feature corresponds to a product and represents whether the user has purchased it or not.
  - 5 different classification tasks - Predict the purchase style

Datasets

- **Type3: Locations**
  - Created from mobile users location “check-ins” in the Foursquare social network
  - Restricted to the Bangkok area For each location venue, geographical position and location type.
  - Partition the Bangkok map
  - 446 binary features, whether a user visited a certain region or location type
  - Classify into 30 different geosocial types

Datasets

- **Type4: Texas hospital stays**
  - Based on the Hospital Discharge Data public use
  - Files with information about inpatient stays in several health facilities
  - Each record contains four main groups of attributes:
    1. External cause of injury (suicide, drug misuse)
    2. Diagnosis (Schizophrenia, illegal abortion)
    3. Underwent procedure of the patient (surgery)
  - Predict the patients main procedure based on the attributes other than secondary procedures

Datasets

- **Type5: MNIST**
  - Handwritten digits
  - Classify numbers 0-10
- **Type6: UCI Adult (Census Income)**
  - People dataset
  - 14 features: age, gender, education, occupation, working hours, native country
  - Binary classification to predict if a person maker over $50K a year

Target models

- Evaluated inference attacks on three types of target models: two constructed by cloud-based "machine learning as a service" platforms and one implemented locally
- **Machine-Learning-as-a-service**:
  - **Google Prediction API**: No configuration parameters
  - **Amazon ML**: control over a few meta-parameters: maximum number of passes, L2 regularization amount
- **Neural Networks**:
  - CIFAR: Convolutional Network
  - Purchase: Fully connected neural network

Experimental Setup

- Training set and test set of each target and shadow model:
  - randomly selected from the respective datasets
  - same size
  - disjoint
- No overlap between the dataset of the target model and those of the shadow models
- Datasets used for different shadow models can overlap with each other.
- Purchase: trained on Google API, Amazon ML & locally employing the same training dataset
  ->Compare leakage from different models

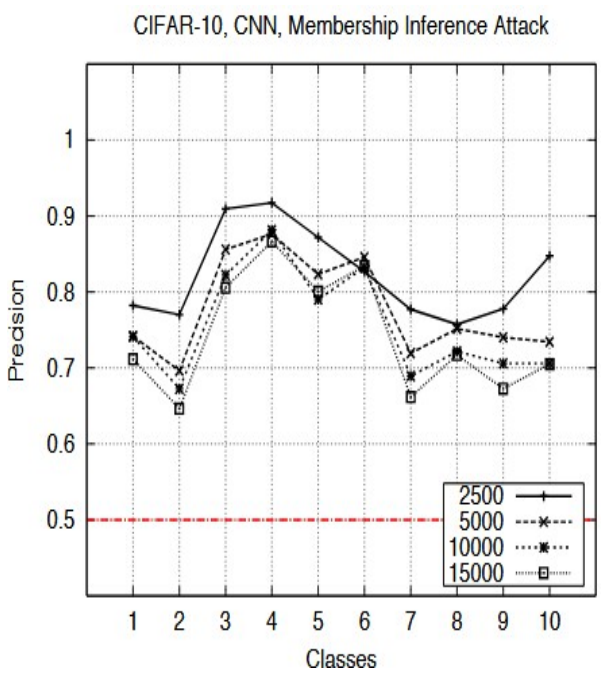
Experimental Setup

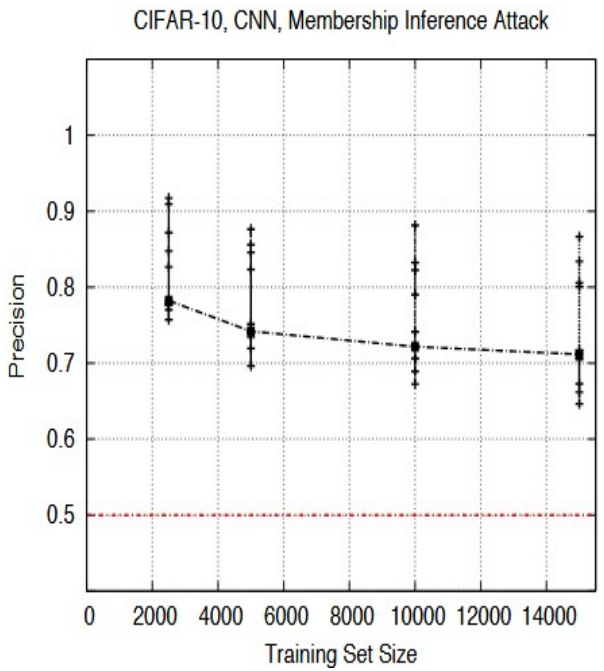
Dataset	Training set size	Cloud/Locally	# classes	# Shadow models
CIFAR	varies*	Locally	10/100	100
Purchases	10,000	Cloud/Locally	{2,10,20,50,100}	20
Locations	1,200	Cloud	30	60
Texas hospital	10,000	Cloud	100	10
MNIST	10,000	Cloud	10	50
UCI Adult	10,000	Cloud	2	20

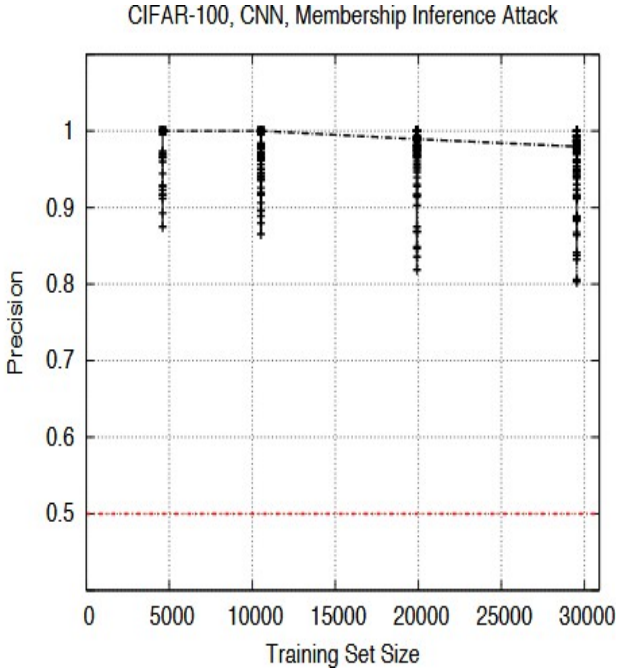
- *Training set sizes
  - CIFAR-10 : 2,500, 5.000, 10,000, 15,000;
  - CIFAR-100: 4,600, 10,520, 19,920, 29,540

Accuracy of the attack

Precision of the membership inference attack against neural networks trained on CIFAR







- The attack performs much better than the baseline, recall is almost 1.0 for both data set

Accuracy of the attack

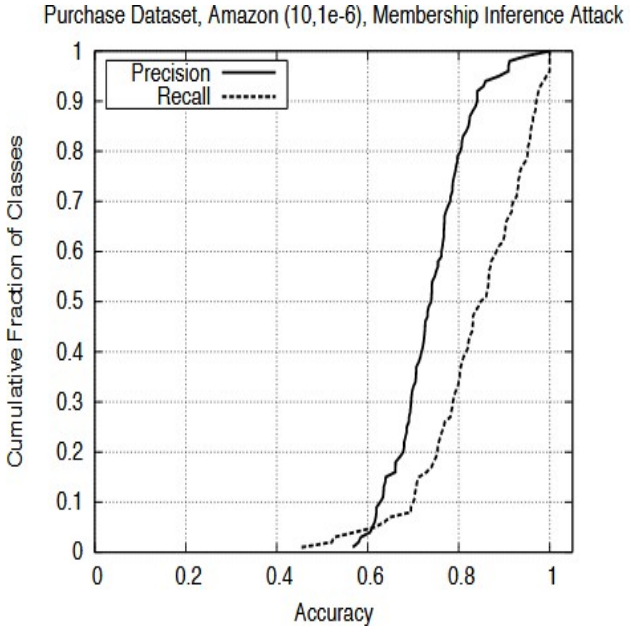
Training & Test accuracy of the models constructed using ML-as-a-service (Purchase dataset 100 classes)

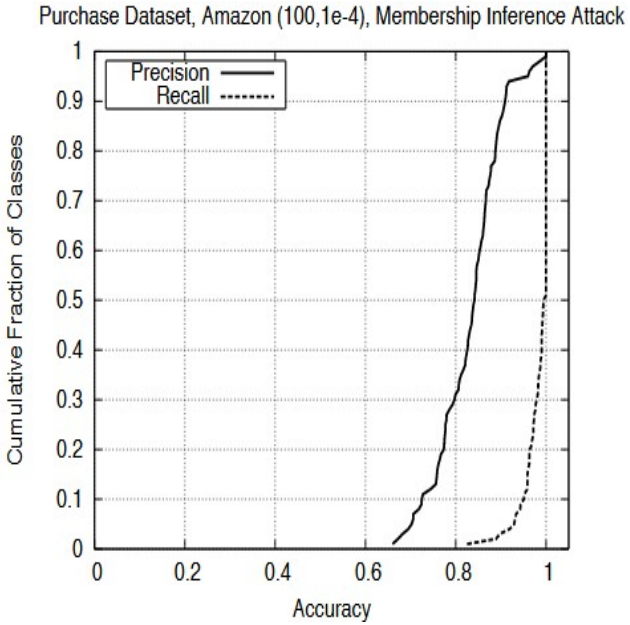
ML Platform	*Training*	*Test*
Google	0.999	0.656
Amazon (10,1e-6)	0.941	0.468
Amazon (100,1e-4)	1.00	0.504
Neural network	0.830	0.670

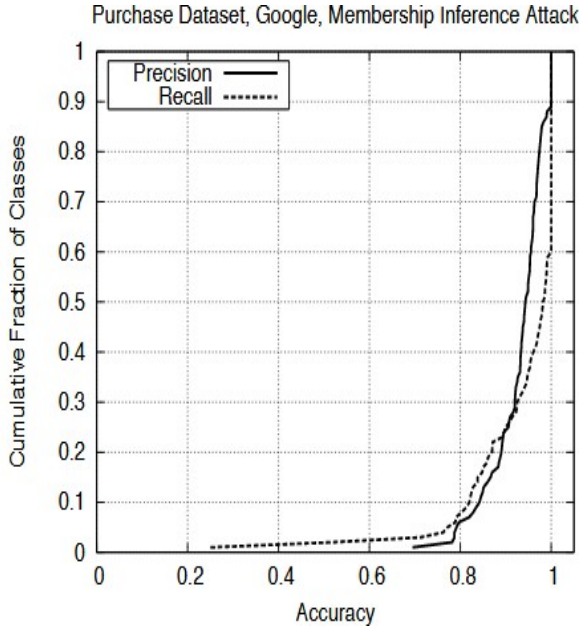
- Large gaps between Training & Test set indicate overfitting
- Larger test accuracy indicates better generalizability and higher predictive power

Accuracy of the attack

- Empirical CDF of the precision and recall of the membership inference attack against different classes of the models trained using Amazon ML and Google API on 10,000 purchase records

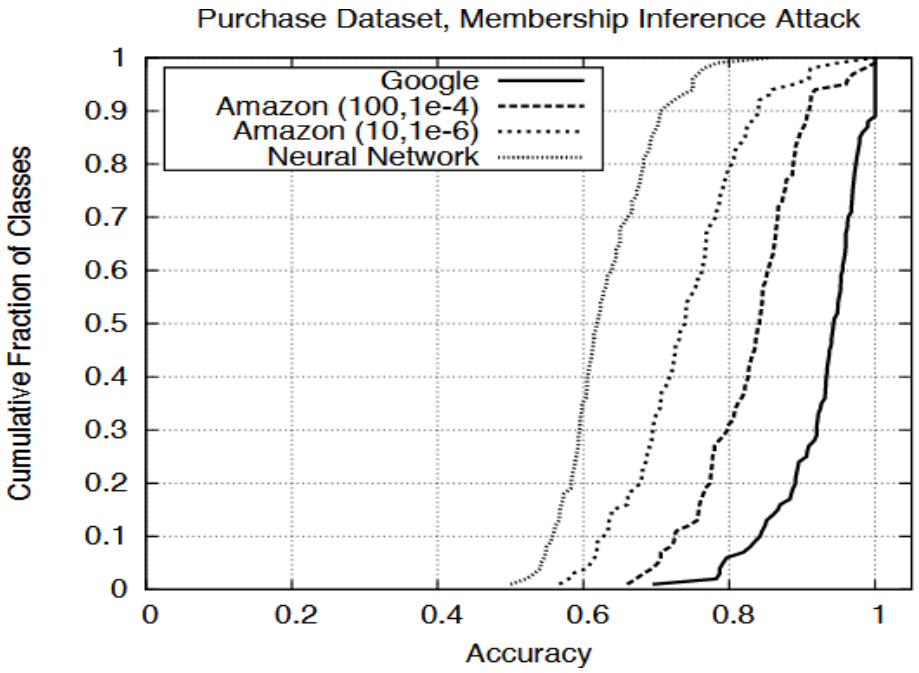






Accuracy of the attack

Precision of MIA against models trained on the same datasets by using different platforms

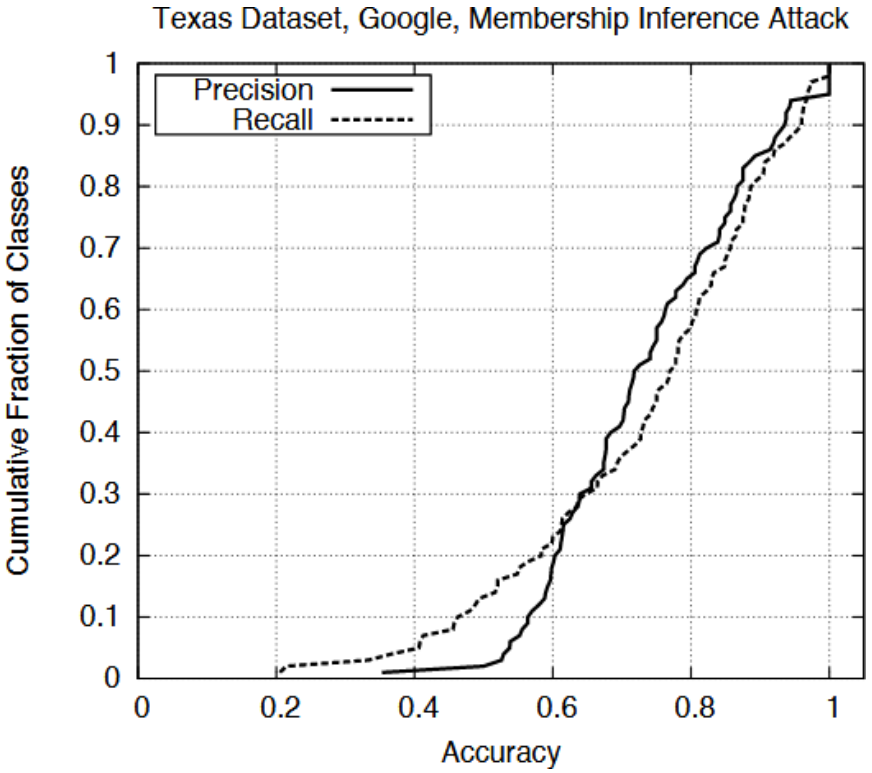


- The attack model is a neural network
- Google Prediction API exhibit the biggest leakage

Accuracy of the attack

Precision and Recall of the MIA against the classification model trained using Google Prediction API (Texas hospital-stay dataset)

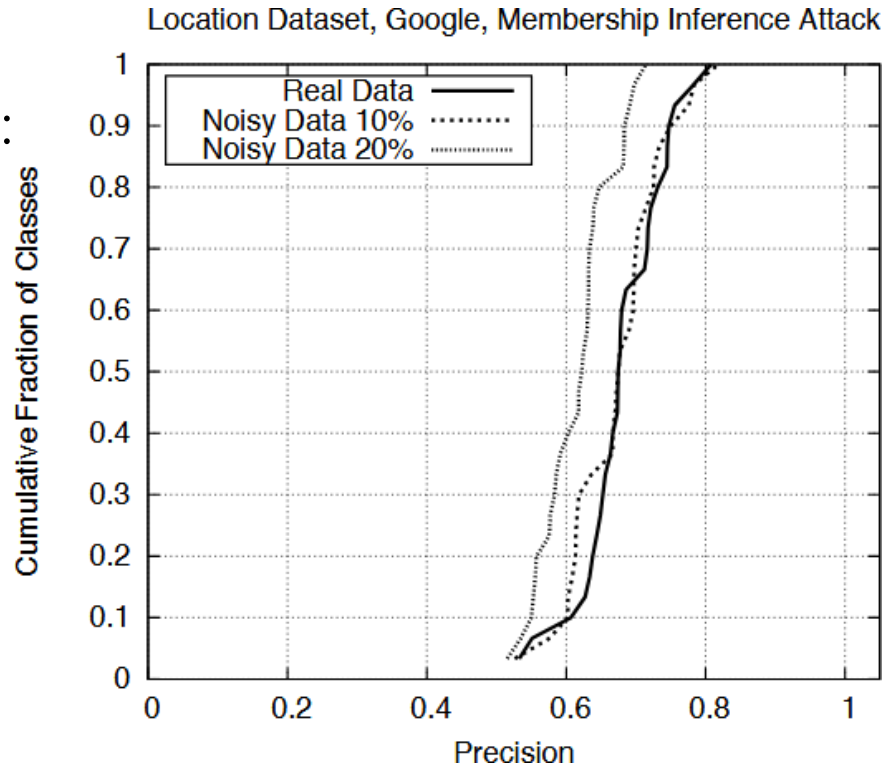
- Training accuracy of the target model: 0.66
- Test accuracy of the target model: 0.51
- Precision is mostly above 0.6
- For half of the classes, precision is above 0.7
- For more than 20 classes precision is above 0.85



Accuracy of the attack

Empirical CDF of the precision of the MAI against the Google-trained model (location dataset)

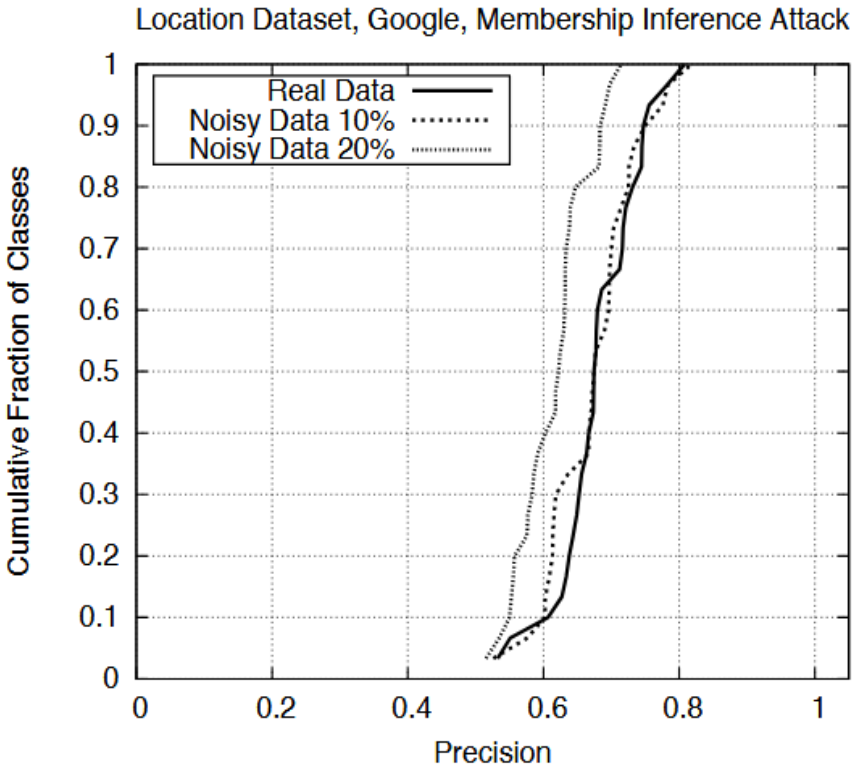
- Training accuracy of the target model: 1.0
- Test accuracy of the target model: 0.66
- Precision is between 0.6 and 0.8 with an almost constant recall of 1.0



Effect of shadow training data with noisy data

Empirical CDF of the precision of the MAI against the Google-trained model (location dataset)

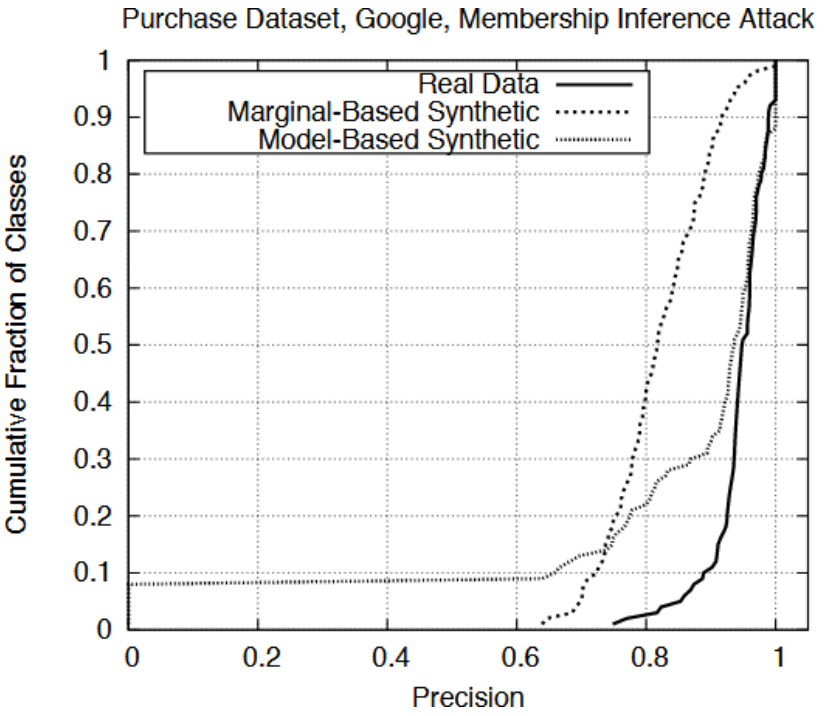
- Precision drops as the amount of noise increases
- The attack outperforms baseline
- **The attacks are robust even if the attacker's assumptions about the distribution of the target models training data are not very accurate**



Effect of shadow training data

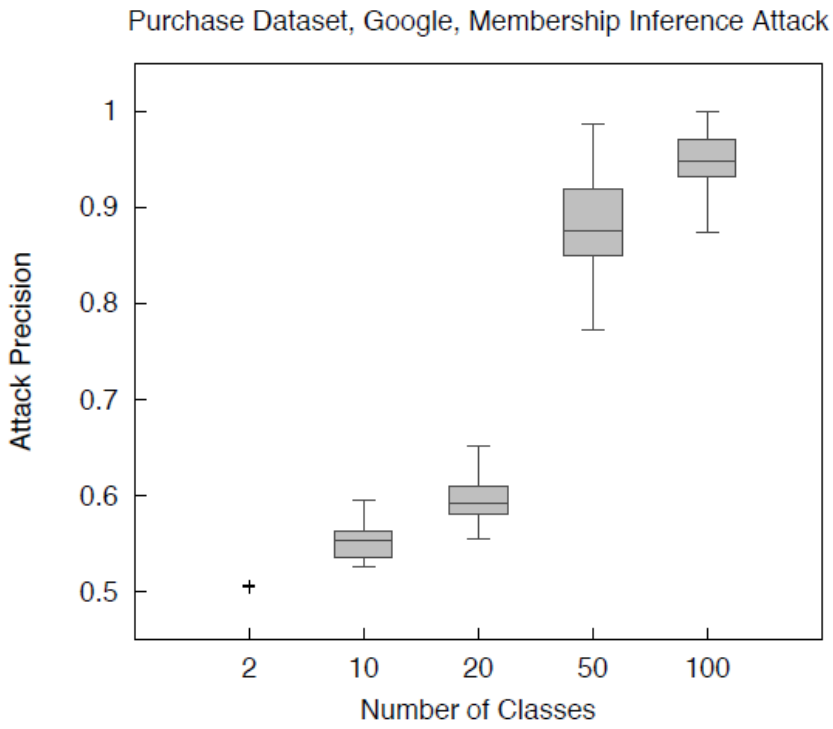
Empirical CDF of the precision of the MIA against the Google-trained model (location dataset)

- The accuracy of the attack using marginal-based synthetic data is noticeably reduced versus real data, but is nevertheless very high for most classes
- a membership inference attack can be trained with only black-box access to the target model, without any prior knowledge about the distribution of the target model's training data



Effect of the number of classes

- Models with fewer classes leak less information about their training inputs
- Models with more output classes need to remember more about their training data, thus they leak more information

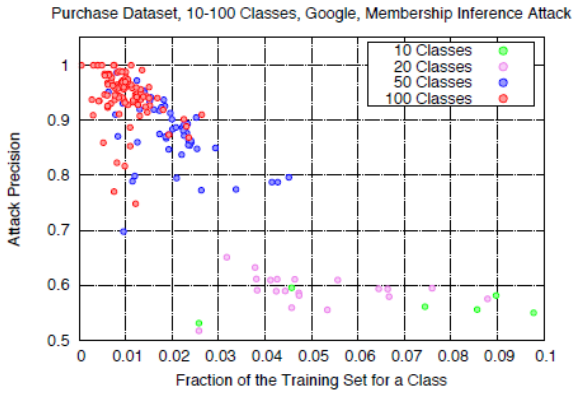


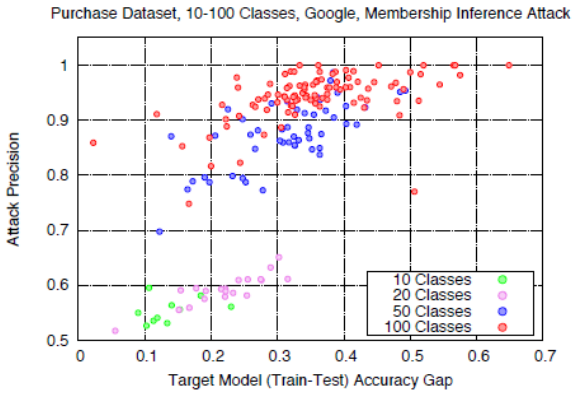
Effect of overfitting

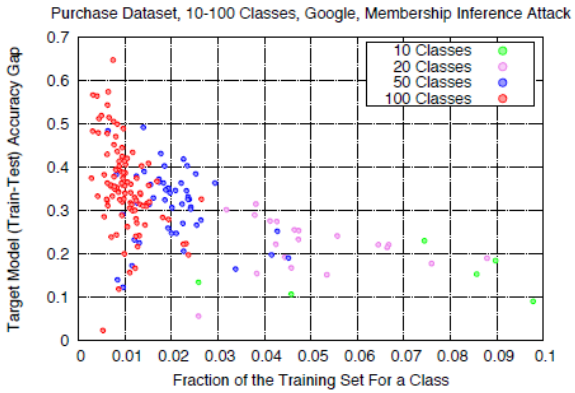
- The more overfitted a model, the more it leaks
- **Overfitting is not the only factor that causes a model to be vulnerable to membership inference**
- The structure and type of the model also contribute to the problem (amazon model vs google model)
- Bigger train-test accuracy gaps indicate that the model is overfitted for that class

Effect of overfitting

- Models with fewer classes leak less information about their training inputs
- High accuracy gap between training data and test data (more overfitting) means more leakage of sensitive information







Why does the attack work?

- Success of membership of inference is directly related to:
  1. Generalizability of the target model
  2. Diversity of its training data

- **Overfitted models lack predictive power and leak sensitive information about the training data**
- If the model overfits and does not generalize well to inputs beyond its training data, or if the training data is not representative, the model leaks information about its training inputs

Mitigation Strategies

- Restrict the prediction vector to top k classes
  - When the number of classes is large, many classes may have very small probabilities in the model's prediction vector
- Coarsen precision of the prediction vector
  - Round the classification probabilities in the prediction vector to d floating point digits The smaller d is, the less information the model leaks

Mitigation Strategies

- Restrict the prediction vector to top k classes
  - When the number of classes is large, many classes may have very small probabilities in the model's prediction vector
- Coarsen precision of the prediction vector
  - Round the classification probabilities in the prediction vector to d floating point digits The smaller d is, the less information the model leaks
- Use regularization
  - L2-norm standard regularization: Penalizes large weight parameters

Pros & Cons

Pros

- Proposed universal membership attack method for services using a machine-learning model
- Proposed shadow training technique that can make attack model for membership inference
- Found that the overfitted model is very weak for training data leakage

Pros & Cons

Cons

- Synthesis procedure works only if the adversary can explore the space of possible inputs
- Synthesis procedure requires many inference steps of the target model
- Actual service models are unlikely to be overfitting
- Some assumptions may not be easy for an attacker
  - Attacker knows the type, architecture, and training algorithm of the target model
  - Background knowledge about the data population
  - Background knowledge about general data population statistics (Marginal distribution of feature values)

Conclusion

- Overfitting is harmful for data leakage
- The attacks are **robust to noisy real data by data generation from the target model**
- **More number of classes** requires the model to store more information -> **leaking more information**

Questions