Monday 22/03/2021

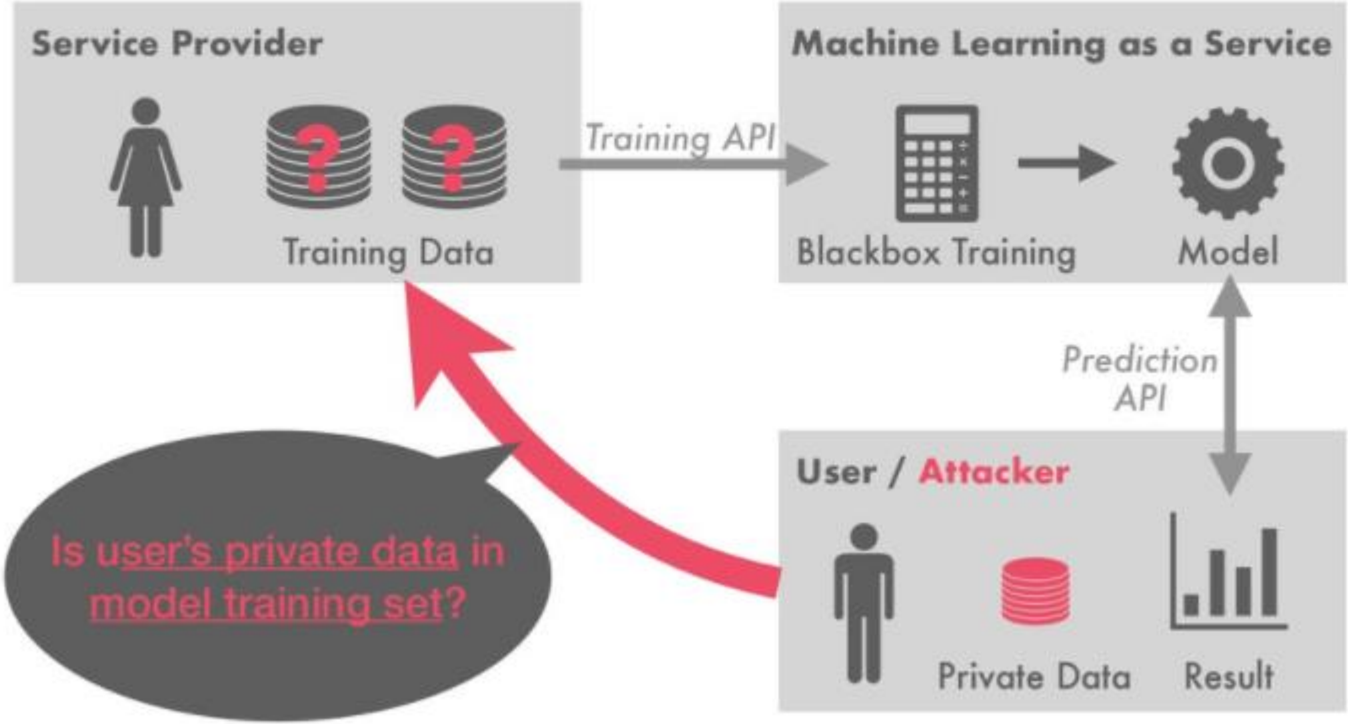
Membership Inference Attacks Against Machine Learning Models

Reza Shokri , Marco Stronati, Congzheng Song, Vitaly Shmatikov

S&P 2017

Contents

Membership inference



https://speakerdeck.com/sorami/acl2020-membership-inference-attacks-on-sequence-to-sequence-models-is-my-data-in-your-machine-translation-system?slide=13

Membership Inference: Consequences

Confidential records and their labels can be identified

- Medical records(disease, past procedures, mental illness, ...)
- Financial records

Paper in a Nutshell

- ➔ Show that you can infer training data of machine learning models (even blackbox!)
- ➔ Propose attack using shadow models trained on synthesized dataset using target model as an oracle
- ➔ Proposed approach can be used to quantify leakage from a specific model

Threat model

- Model trained on private data can be released and leak training samples
  - Commercial models trained on large training sets
  - Tailored models

- Some datasets are sensitive

Proposed Attack

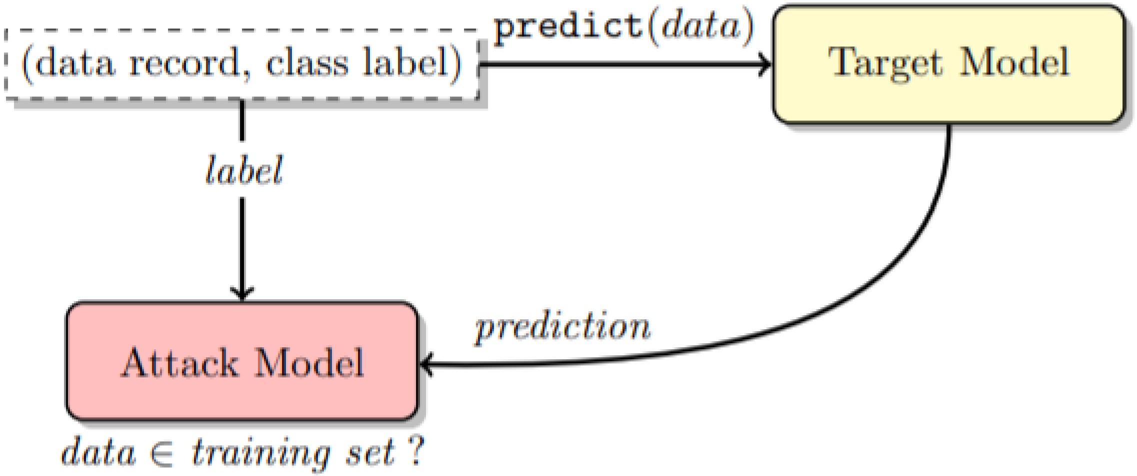
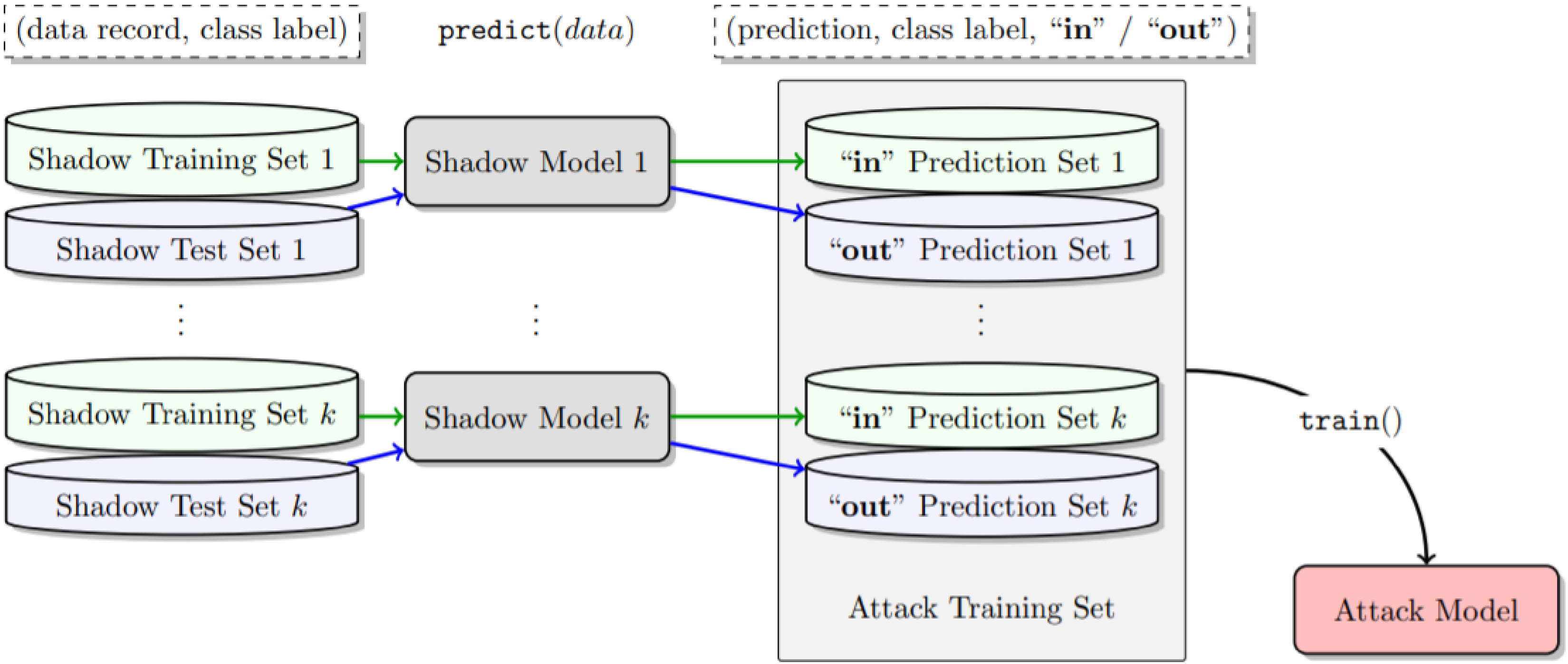


Figure 1

Attack

1. Generate data points that maximise target mode prediction confidence
2. Train shadow models on generated dataset
3. Given shadow models prediction output, create attack training dataset
4. Train attacker model on shadow dataset

Attack


(data record, class label)
predict(data)
(prediction, class label, "in" / "out")
Shadow Training Set 1
Shadow Test Set 1
Shadow Model 1
"in" Prediction Set 1
"out" Prediction Set 1
Shadow Training Set k
Shadow Test Set k
Shadow Model k
"in" Prediction Set k
"out" Prediction Set k
Attack Training Set
train()
Attack Model

Shadow model dataset generation

1. Synthetic data
   - ➔ Generate synthetic data that is classified with high accuracy by target model (through queries)
   - ➔ Hill climbing
   - ➔ Doesn’t work on all type of inputs (high resolution pictures)

2. Noisy data
   - ➔ Replace some values with random values from original dataset

3. Statistic based:
   - ➔ Sample from marginal distribution over features

Algorithm 1 Data synthesis using the target model

procedure SYNTHESIZE(class : c)
    $\mathbf{x} \leftarrow$ RANDRECORD( ) $\triangleright$ *initialize a record randomly*
    $y_c^* \leftarrow 0$
    $j \leftarrow 0$
    $k \leftarrow k_{max}$
    for $iteration = 1 \cdots iter_{max}$ **do**
        $\mathbf{y} \leftarrow f_{\text{target}}(\mathbf{x})$ $\triangleright$ *query the target model*
        if $y_c \geq y_c^*$ **then** $\triangleright$ *accept the record*
            if $y_c > \text{conf}_{min}$ and $c = \arg\max(\mathbf{y})$ **then**
                if $\text{rand}() < y_c$ **then** $\triangleright$ *sample*
                    return $\mathbf{x}$ $\triangleright$ *synthetic data*
                end if
            end if
            $\mathbf{x}^* \leftarrow \mathbf{x}$
            $y_c^* \leftarrow y_c$
            $j \leftarrow 0$
        else
            $j \leftarrow j + 1$
            if $j > rej_{max}$ **then** $\triangleright$ *many consecutive rejects*
                $k \leftarrow \max(k_{min}, \lceil k/2 \rceil)$
                $j \leftarrow 0$
            end if
        end if
        $\mathbf{x} \leftarrow$ RANDRECORD($\mathbf{x}^*$, k) $\triangleright$ *randomize k features*
    end for
    return $\perp$ $\triangleright$ *failed to synthesize*
end procedure

Phase 1: Shadow models

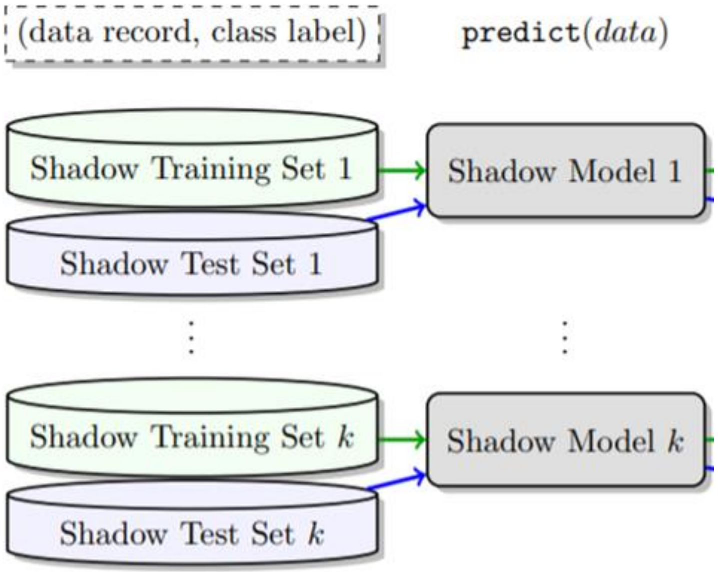
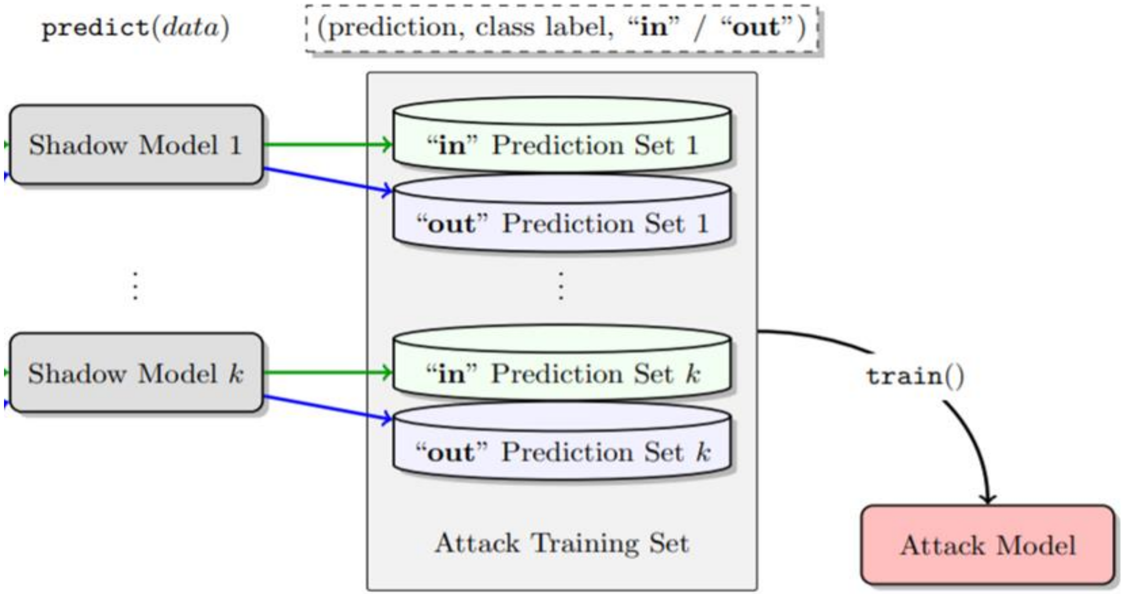


Figure 1

- Train model with similar architectures as target model
    - Shadow model
- Outputs probability and class

Phase 2: Attacker model



- Attacker takes shadow models prediction probabilities over the classes
- Training set labelled as **IN**
- Test set labeled as **OUT**
- For each class, train an attacker model

Experiments

→ Attack precision

→ Training set size

→ Number of classes

→ Different data sampling

Experiment setup

- ➔ Split shadow sets for IN and OUT samples
- ➔ Train cloud-based models
- ➔ Train shadow models
- ➔ Evaluate attacker model

Datasets and Tasks

- **CIFAR10 & CIFAR 100:** image recognition
- **Purchases**: Predict the purchase style (2, 10, 20, 50, 100 classes)
- **Locations**: Predict the user's geosocial type given their record (30 classes)
- **Texas hospital stays**: Classify patient procedure (100 classes)
- **MNIST:** handwritten digits recognition (10 classes)
- **UCI Adult:** Census data for binary income classification using age, gender, education, marital status, occupation, working hours, native country.

Target models
Blackbox

Target:

- ➔ Google Prediction API
- ➔ Amazon ML :
  - ◆ Model with 10 max passes and L2 = 1e-6
  - ◆ Model with 100 max passes and L2 = 1e-4
- ➔ **Local**: NN, CNN

Shadows :

- ➔ NN
- ➔ CNN

Training

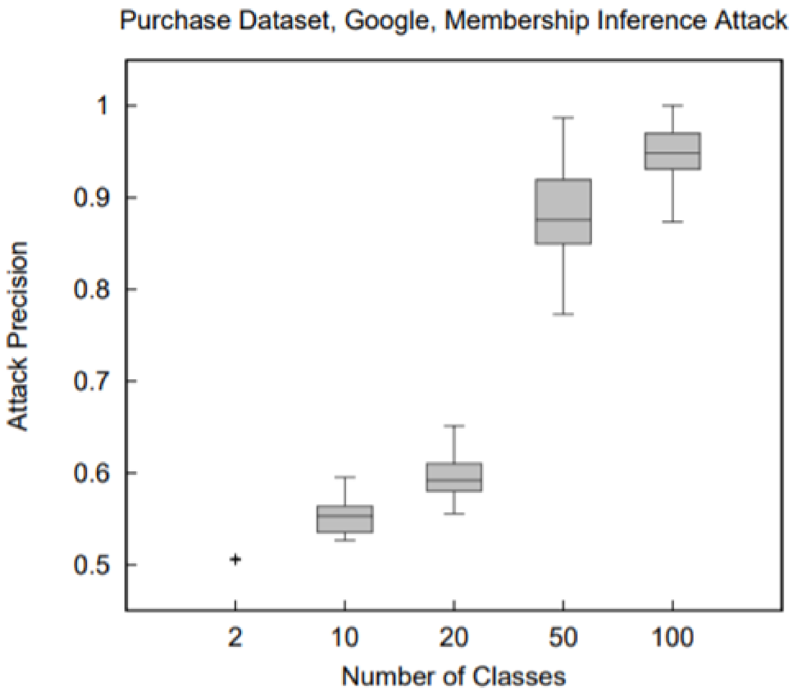
- ➔ Split dataset between target models and shadow models
- ➔ Shadow models datasets can overlap
- ➔ 10 000 samples for all sets expect for Locations with 1,200 samples
- ➔ Shadow models count:
  - ◆ CIFAR: 100
  - ◆ Purchase: 20
  - ◆ Texas hospital: 10
  - ◆ Location: 60
  - ◆ MNIST: 50
  - ◆ Census: 20

Results

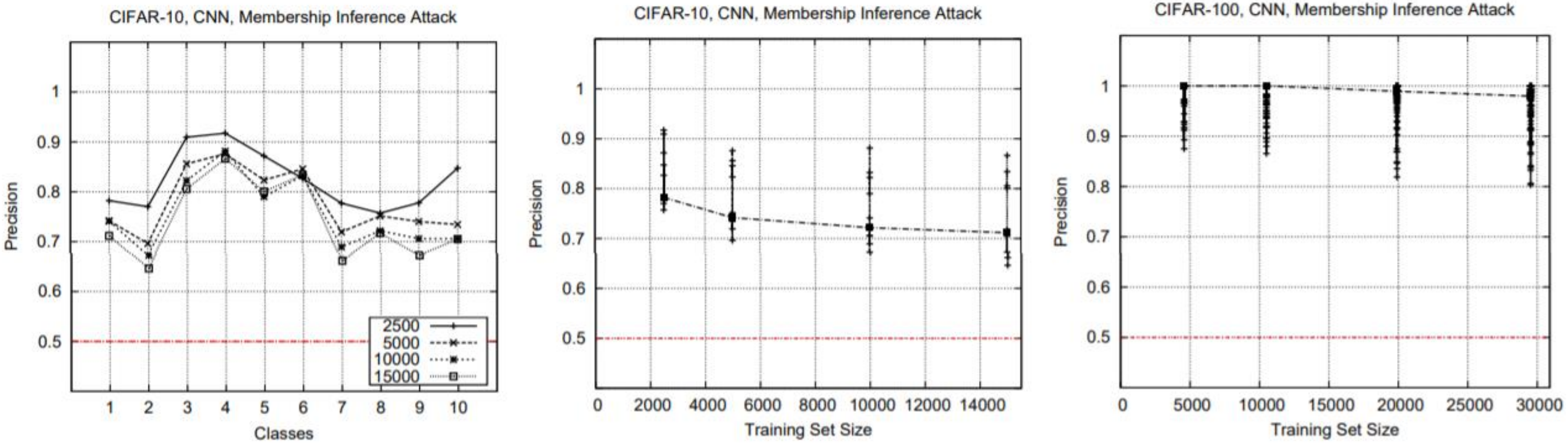
Attack Precision (Google)

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

Number of classes (Google)

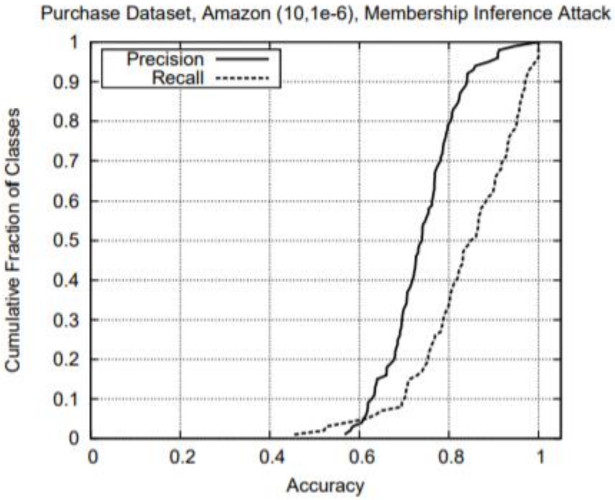


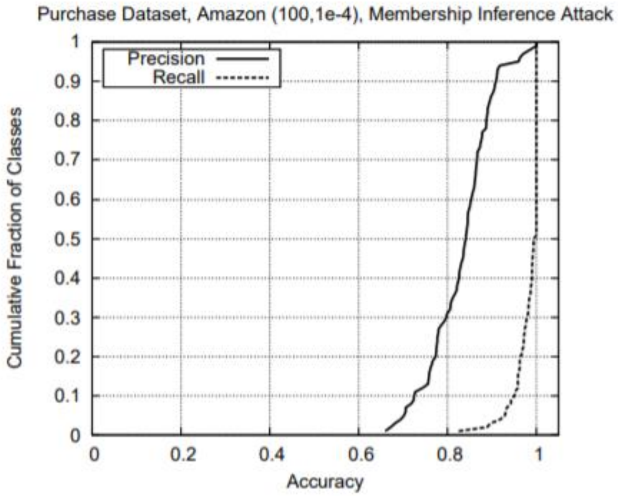
Effect of classes & training set size on CIFAR (Local)

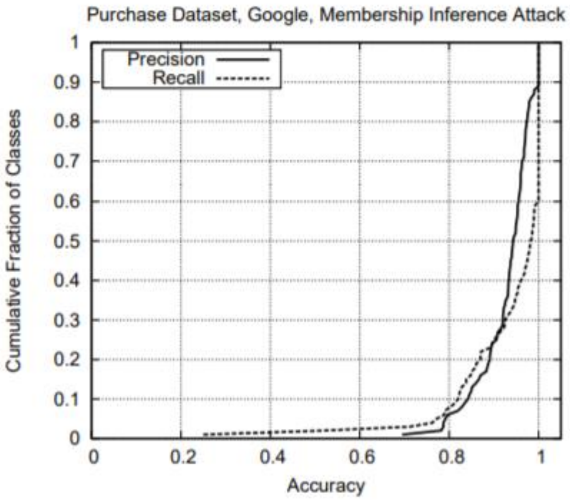


Precision and Recall over classes

Purchase dataset (30 classes)

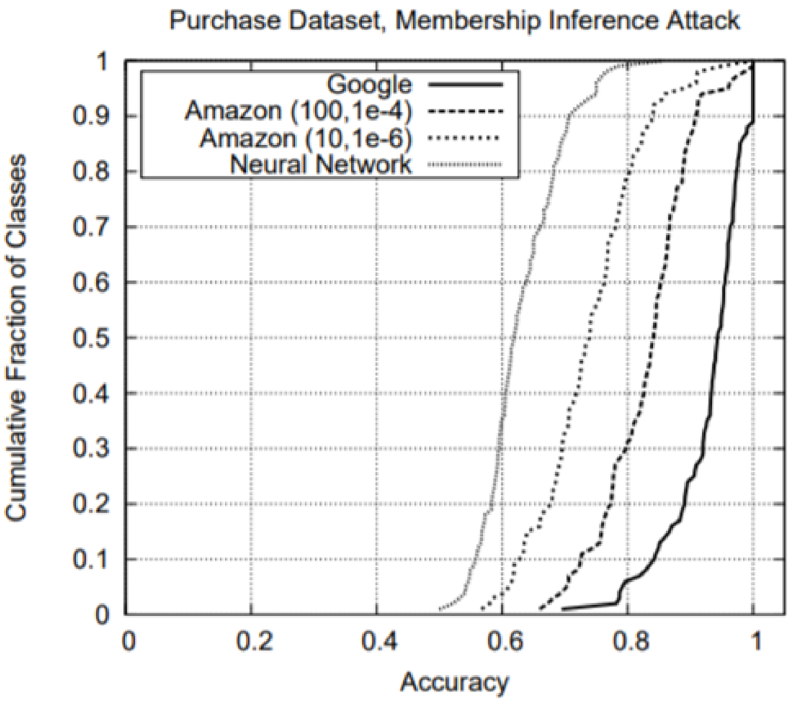




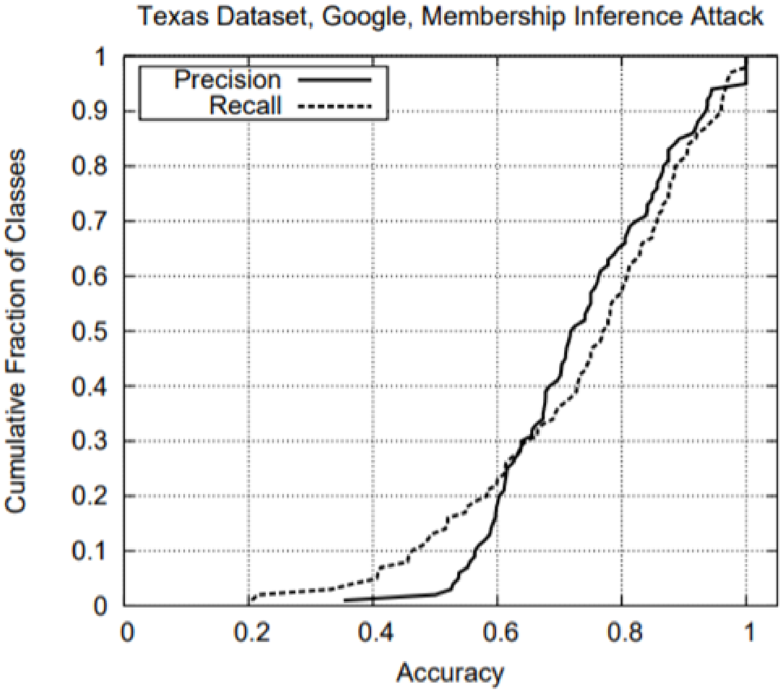


Results on Purchase dataset (30 classes)

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

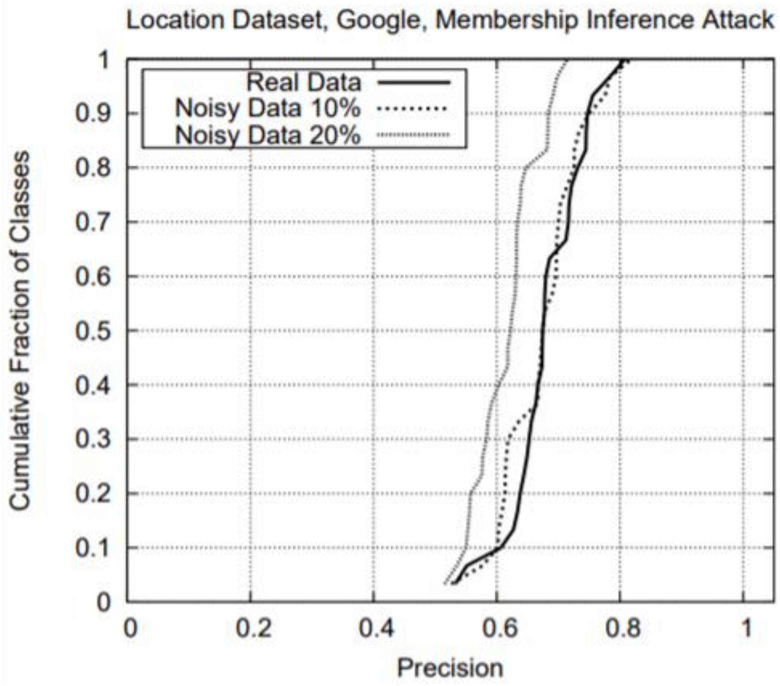


Results on Texas Hospital dataset (100 classes)



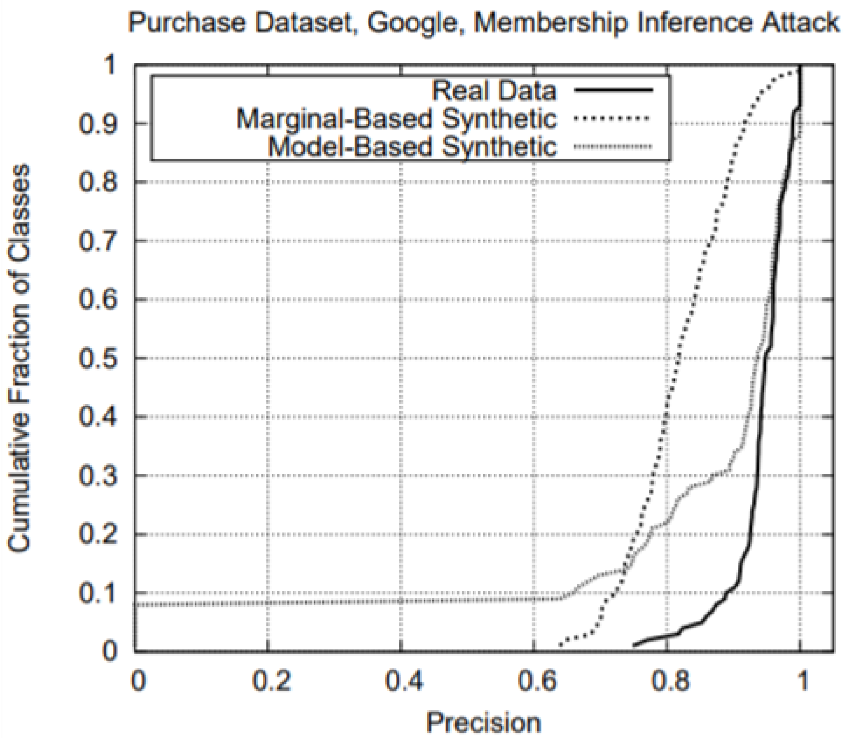
- Training accuracy: 0.66
- Test accuracy: 0.51

Noisy data
Location dataset



- Precision of the attack over all classes is 0.678 (real data),
- 0.666 (data with 10% noise), and 0.613 (data with 20% noise).
- The corresponding recall of the attack is 0.98, 0.99, and 1.00, respectively
- The training accuracy of the target model is 1 and its test accuracy is 0.66.

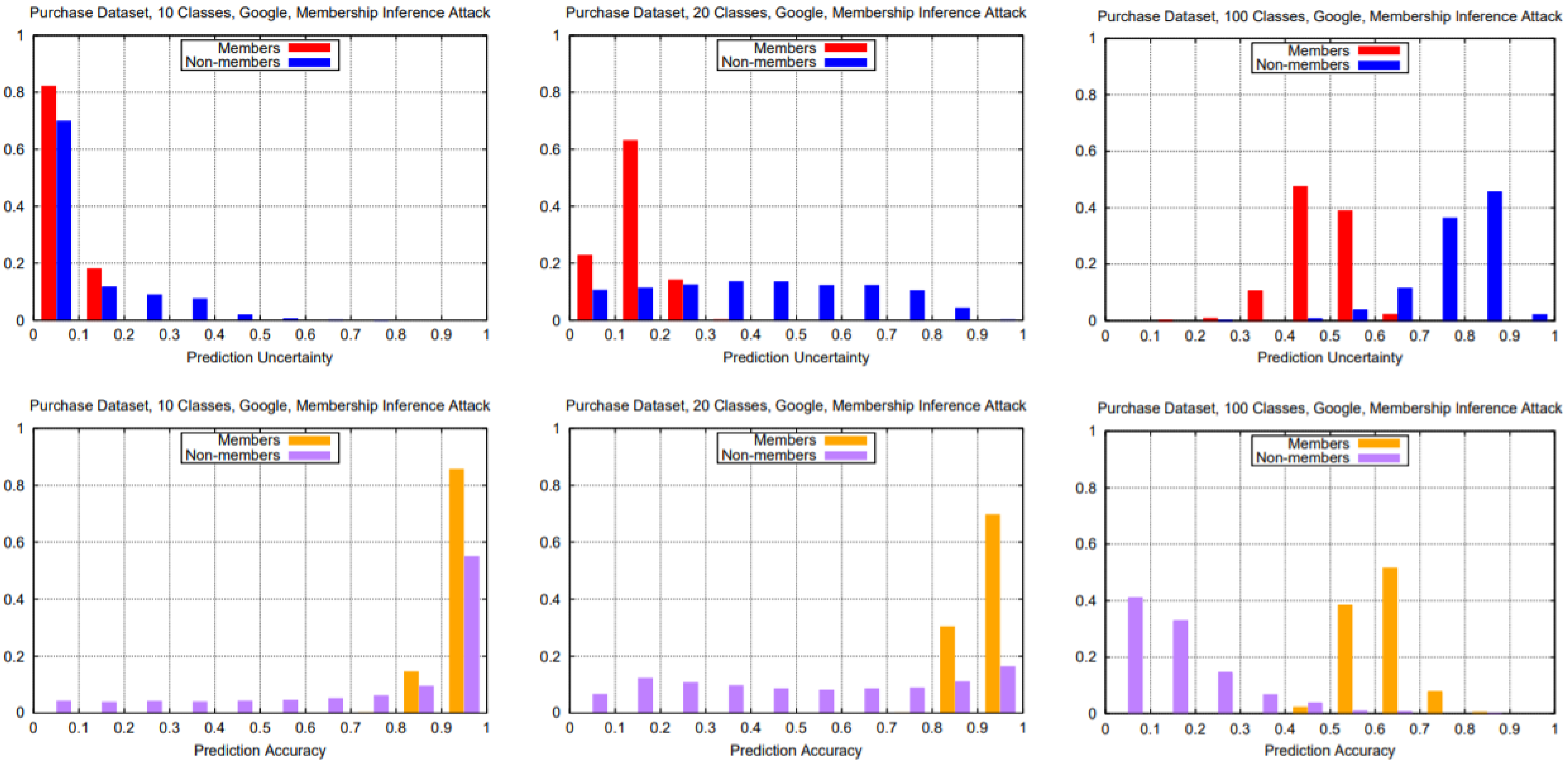
Marginal data



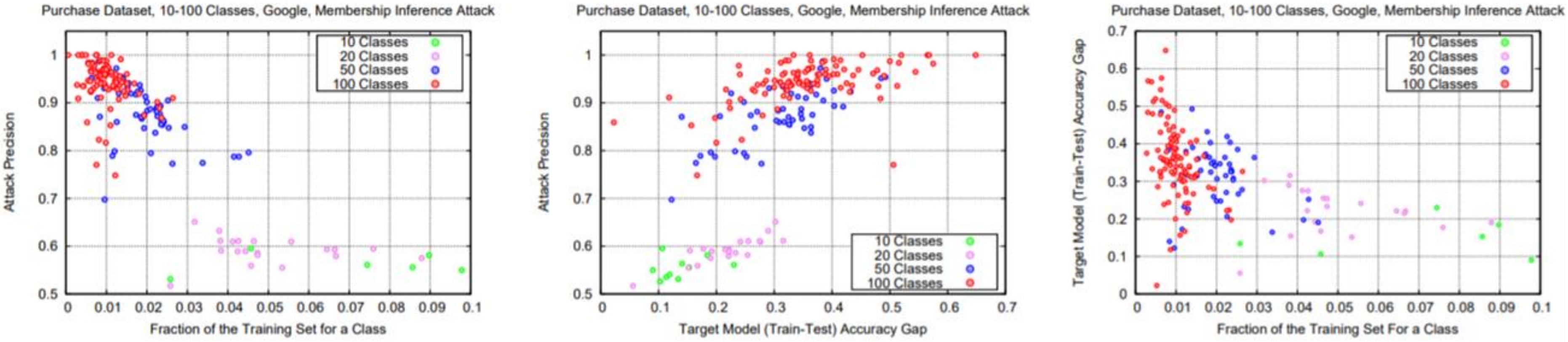
- Precision of the attack over all classes is 0.935 (real data)
- 0.795 (marginal-based synthetic data)
- 0.896 (model-based synthetic data).
- The corresponding recall of the attack is 0.994, 0.991, and 0.526, respectively.

Why?

- ➔ Exploit the overfitting
- ➔ Model outputs probability with high confidence on training data than on test set


Purchase Dataset, 10 Classes, Google, Membership Inference Attack
Members
Non-members
Prediction Uncertainty
Purchase Dataset, 20 Classes, Google, Membership Inference Attack
Members
Non-members
Prediction Uncertainty
Purchase Dataset, 100 Classes, Google, Membership Inference Attack
Members
Non-members
Prediction Uncertainty
Purchase Dataset, 10 Classes, Google, Membership Inference Attack
Members
Non-members
Prediction Accuracy
Purchase Dataset, 20 Classes, Google, Membership Inference Attack
Members
Non-members
Prediction Accuracy
Purchase Dataset, 100 Classes, Google, Membership Inference Attack
Members
Non-members
Prediction Accuracy

Overfitting



Discussion

1. Higher number of classes increases the attack accuracy
2. The attack accuracy is reduced
   a. for classes with less samples
   b. For larger training sets
3. Overfitting is the main cause of attack success

1. Other use cases:
   a. License breach
   b. Dataset for challenges

Mitigation

Mitigation Strategies

1. Return only top k classes probabilities
2. Round class probabilities
3. Increase entropy of the prediction vector
    - Increase normalizing temperature in softmax
4. Add regularization to loss function

Mitigation Strategies: Evaluation

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

Hospital dataset	*Testing Accuracy*	*Attack Total Accuracy*	*Attack Precision*	*Attack Recall*
No Mitigation	0.55	0.83	0.77	0.95
Top $k = 3$	0.55	0.83	0.77	0.95
Top $k = 1$	0.55	0.82	0.76	0.95
Top $k = 1$ label	0.55	0.73	0.67	0.93
Rounding $d = 3$	0.55	0.83	0.77	0.95
Rounding $d = 1$	0.55	0.81	0.75	0.96
Temperature $t = 5$	0.55	0.79	0.77	0.83
Temperature $t = 20$	0.55	0.76	0.76	0.76
L2 $\lambda = 1e-4$	0.56	0.80	0.74	0.92
L2 $\lambda = 5e-4$	0.57	0.73	0.69	0.86
L2 $\lambda = 1e-3$	0.56	0.66	0.64	0.73
L2 $\lambda = 5e-3$	0.35	0.52	0.52	0.53

Mitigation Strategies: Evaluation

1. Even for restricting to one class is not enough
2. Attack can exploit the mislabeling behavior
3. Regularization is beneficial for model generalisability and defense against inference attack
4. Not all methods can be implemented in practice
    - High temperature reduces model accuracy

Mitigation

- Reduce overfitting :
  - Dropout
  - Batch normalisation
- Reduce model complexity
- Differential private training
- Avoid small datasets
- Warn users about risks

Limitations

1. Assume target models outputs probability over classes
    a. Sequence to sequence models?
    b. ASR ?
2. How to choose optimal number of shadow models?
3. Inconsistent comparisons
4. Consequent work show that thresholding is sufficient [1]

[1] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In IEEE Computer Security Foundations Symposium (CSF). 268–282.

Conclusion

1. Paper trains shadows models on generated data labelled by target model to perform membership inference on blackbox models

1. Shows that overfitting leads to data leakage

1. Most effective mitigation strategy is regularization

Related Work

1. **Unintentional memorisation for generative text models**
    a. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks, Carlini et al., Usenix 19’
2. **Follow up Defense with Adversarial model:**
    a. Machine Learning with Membership Privacy using Adversarial Regularization, Milad et al., ACM SIGSAC Conference on Computer and Communications Security, 2018.
3. **Speech**:
    a. The Audio Auditor: User-Level Membership Inference in Internet of Things Voice Services, Miao et al, 2019

Thank you!