The Audio Auditor: User-Level Membership Inference in Internet of Things Voice Services

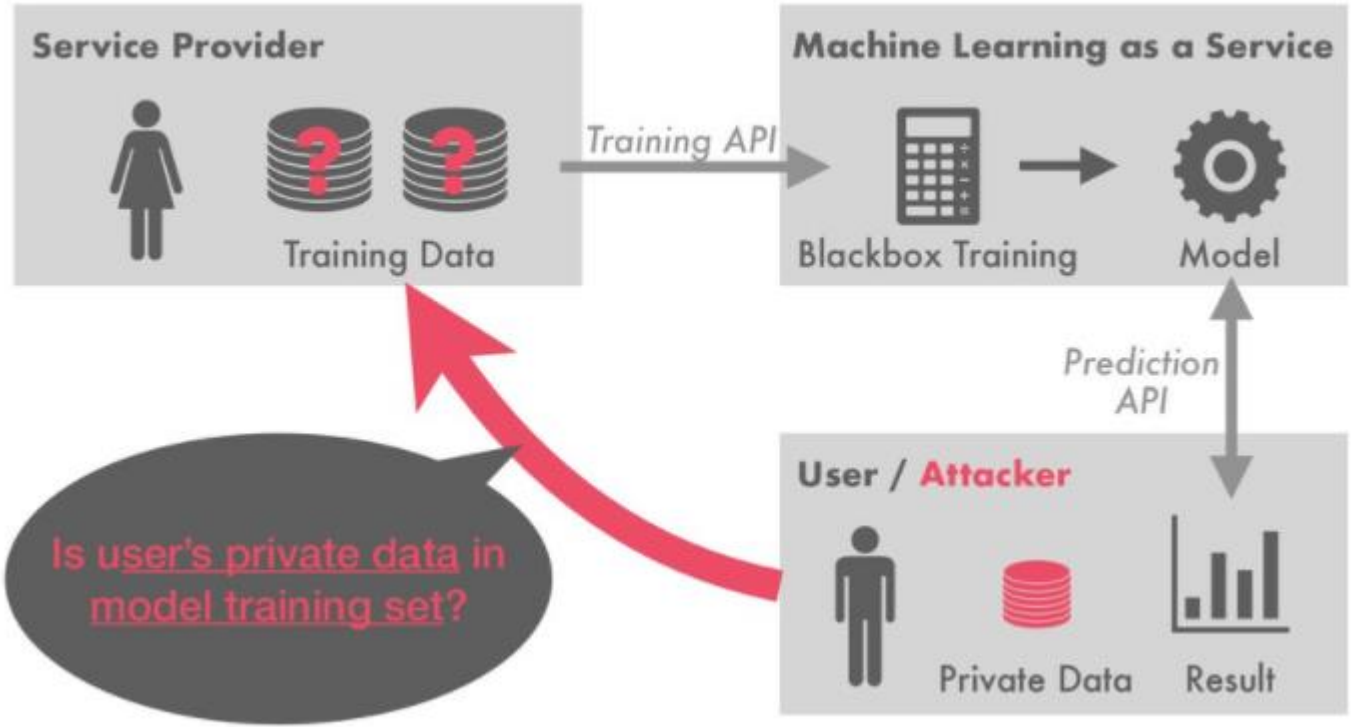
Yuantian Miao, Minhui Xue, Chao Chen, Lei Pan, Jun Zhang, Benjamin Zi Hao Zhao, Dali Kaafar, Yang Xiang

PETS 2021

Contents

- User membership Inference
- Audio Auditor
- Efficacity, Efficiency, Transferability and robustness of attack
- Discussion

Record-level 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

Record-level Membership inference

- ➔ Single record
- ➔ Check if that specific record is in the training set
- ➔ Works best for tabular data

User-level Membership inference

- ➔ Multiple records per user
- ➔ Check if any sample of user was part of training set
- ➔ Speech recording, photos, texts...

Paper in a Nutshell

Proposed approach allow user to see if their data was used for the training of a voice assistant

- ➔ Perform user membership inference on black box ASR systems
- ➔ Build shadow models and auditor based on transcriptions and audio features
- ➔ Demonstrate attack on Apple's Siri

Previous work: Membership Inference Attacks Against Machine Learning Models

- ➔ Paper trains shadows models on generated data labelled by target model to perform membership inference on blackbox models
- ➔ Uses output probability
- ➔ Shows that overfitting leads to data leakage
- ➔ Most effective mitigation strategy is regularization

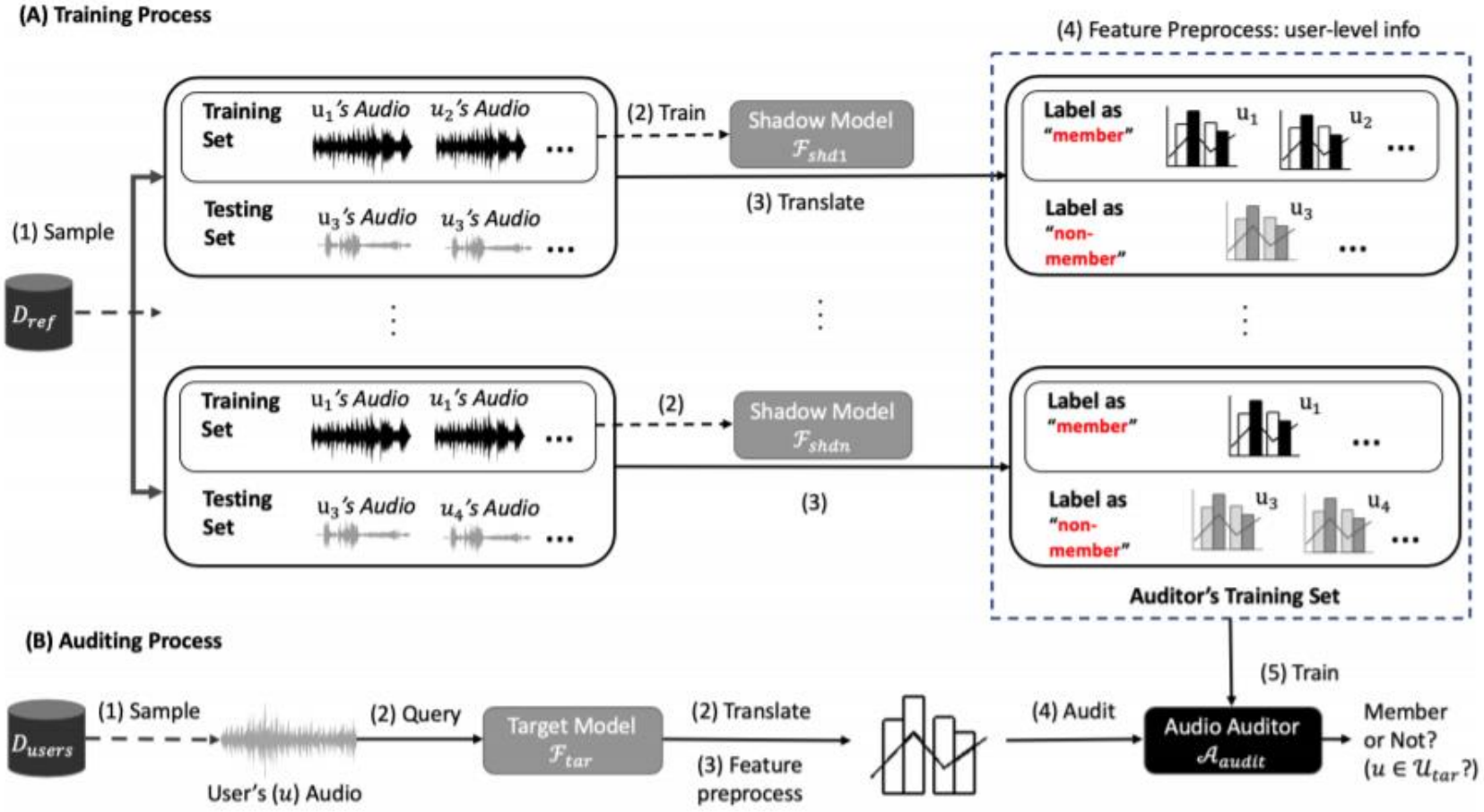
Threat model

Model trained on private data can be released and leak training samples

- Commercial models trained on large training sets
- Voice cloning
- Some manufacturers might not respect privacy policies

Assumptions:

- Black box models
- Access to transcription only
- Auditor queries black box models to identify membership of user in training set


(A) Training Process
(4) Feature Preprocess: user-level info
Training Set
u_1's Audio
u_2's Audio
(2) Train
Shadow Model $\mathcal{F}_{shd1}$
Testing Set
u_3's Audio
u_3's Audio
(3) Translate
(1) Sample
D_{ref}
Label as "member"
u_1
u_2
Label as "non-member"
u_3
Training Set
u_1's Audio
u_1's Audio
(2)
Shadow Model $\mathcal{F}_{shdn}$
Testing Set
u_3's Audio
u_4's Audio
(3)
Label as "member"
u_1
Label as "non-member"
u_3
u_4
Auditor's Training Set
(B) Auditing Process
(5) Train
D_{users}
(1) Sample
User's (u) Audio
(2) Query
Target Model $\mathcal{F}_{tar}$
(2) Translate
(3) Feature preprocess
(4) Audit
Audio Auditor $\mathcal{A}_{audit}$
Member or Not? ($u \in \mathcal{U}_{tar}$?)

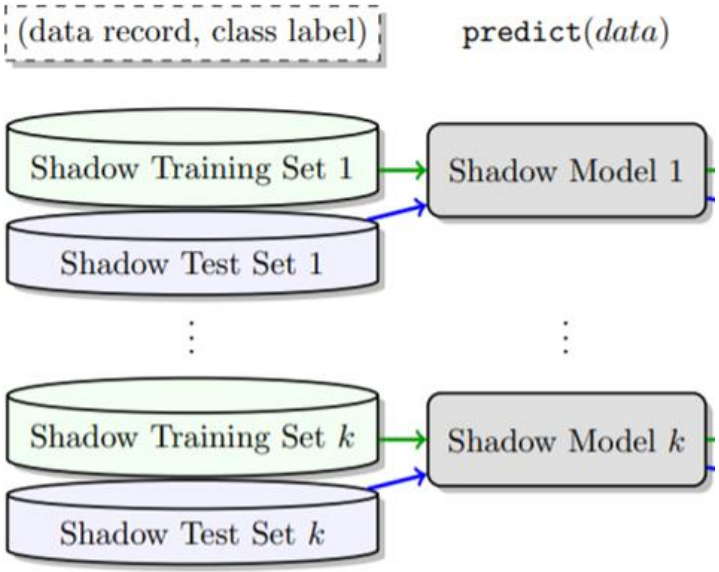
Methodology

1. Training phase
    a. Train ASR shadow models
2. Auditor phase
    a. Generate transcriptions with shadow model
    b. Extract features
    c. Train an auditor on the extracted features

Features

1. Extract speaker speed and frame length
2. Source transcription to compare with target transcription
    a. Cosine similarity between transcription_src and transcription_target
    b. Missing characters and insertions
3. Compute mean, std, median, max, min and variance per user

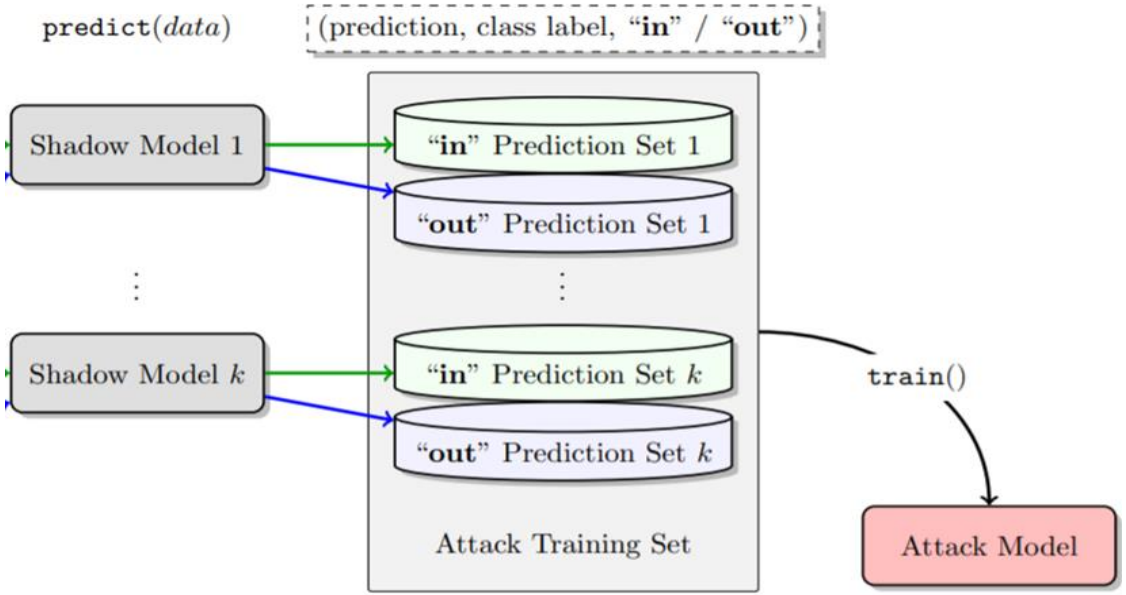
Phase 1: Shadow models



- Train model with similar architectures as target model
  - Shadow model
- Outputs transcription

Figure 1

Phase 2: Attacker model



- Attacker takes transcription and extract features
- Training set labelled as **IN**
- Test set labeled as **OUT**
- One attacker model

Experiments

➔ **RQ1**: How performant is the auditor and how does it perform with different size of training sets?

➔ **RQ2:** How many queries are necessary?

➔ **RQ3:** Transferability across datasets

➔ **RQ4:** Robustness to different blackbox models

➔ Real-world implementation

Experiments setup

- ➔ Split shadow sets for IN and OUT samples
- ➔ Train ASR target model
- ➔ Train ASR shadow model
- ➔ Train auditor on shadow training set
- ➔ Evaluate attacker model (Accuracy, Precision and Recall)

Experiments setup

Assumptions:

- ➔ Target ASR and Shadow ASR have same pipeline (Hybrid ASR)
  - ◆ Target ASR: LSTM
  - ◆ Shadow ASR: GRU
- ➔ Auditor model is a random forest
- ➔ Train Target ASR and shadow ASR on Librispeech

RQ1: Auditor Performance (wr training size)

Experiment

- ➔ Build a shadow set using N users.
- ➔ Examine extreme cases:
  - ◆ Auditor’s testing set only contains member users querying with the unseen audios
  - ◆ Auditor’s testing set that only contains member users querying with the seen audios
- ➔ Experiment with number of samples per user in the shadow set

Average performance and extremes cases

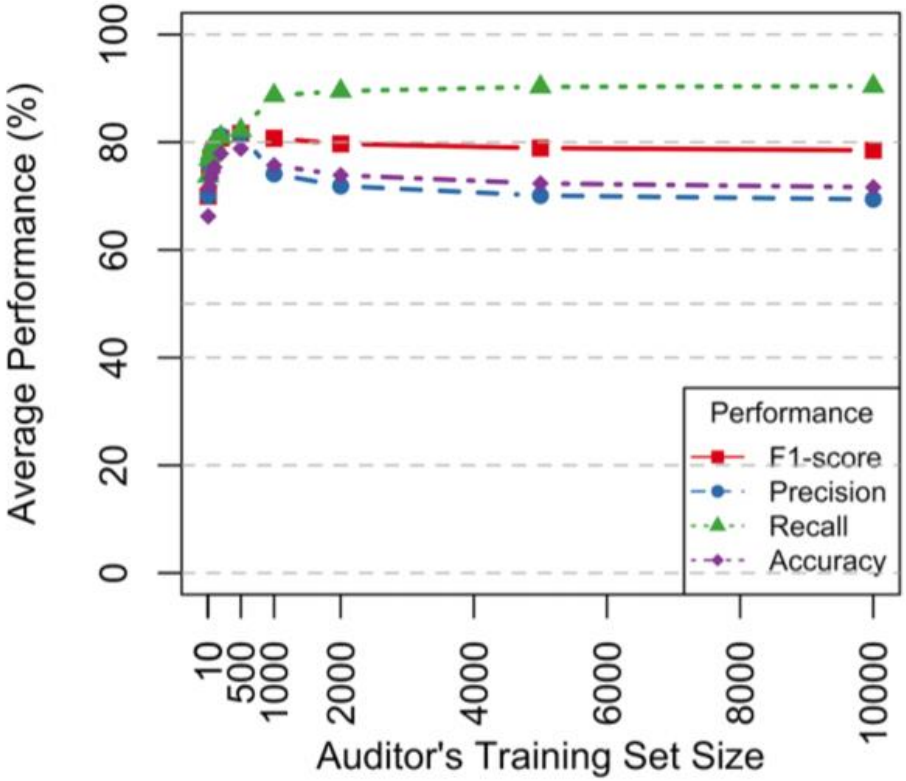


Fig. 3. Auditor model performance with varied training set size.

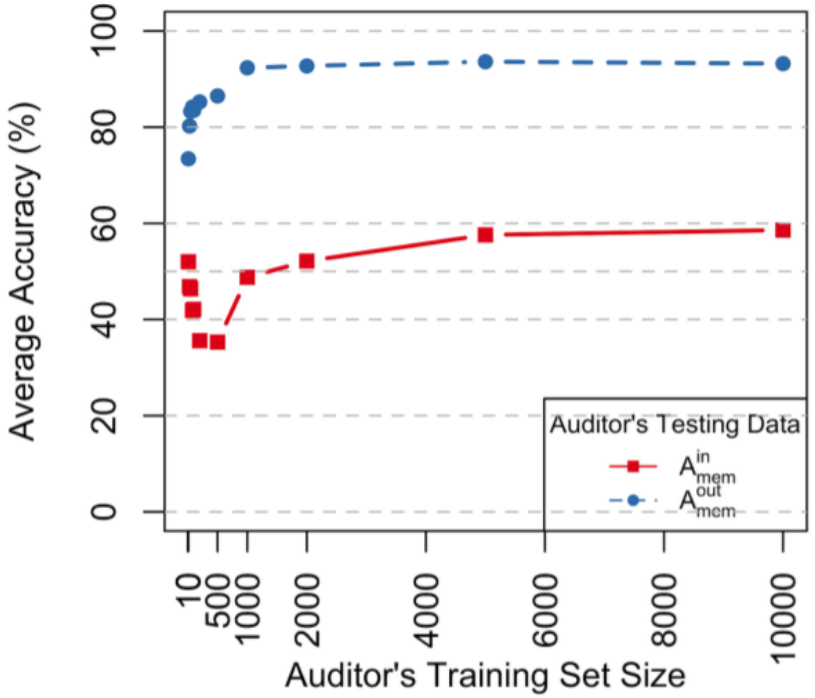


Fig. 4. Auditor model accuracy on a member user querying with the target model's unseen audios (A^{out}_{mem}) against the performances on the member users only querying with the seen recordings (A^{in}_{mem}).

Impact of samples size per user

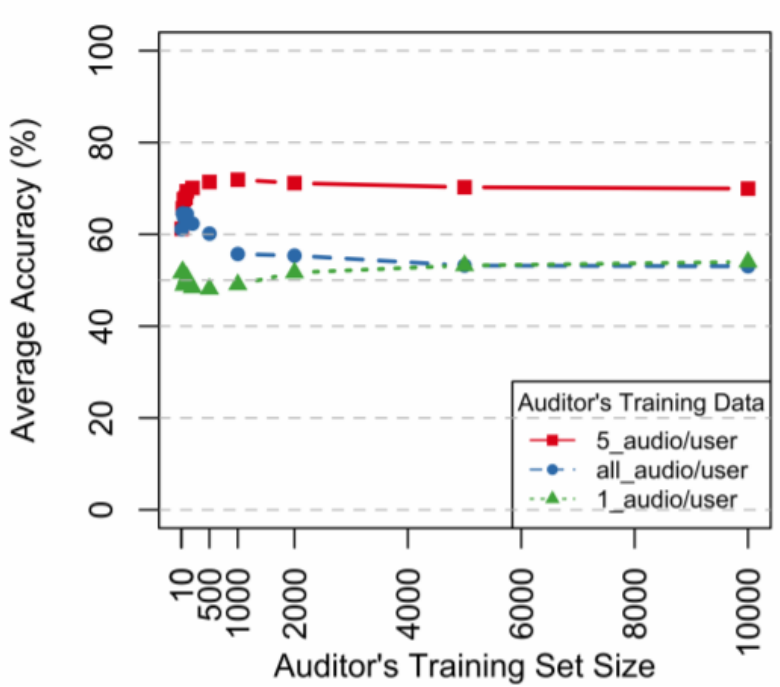


Fig.6. A comparison of average accuracy for one audio, five audios, and all audios per user when training the auditor model with a limited number of audios per user gained in the auditing phase.

Impact of number of users

1. The auditor performs best on samples seen during the target ASR training
2. Larger training size lower the precision
3. The performance peak around 5000 users
4. When training the auditor, 5 samples per users is optimal

RQ2: Number of queries

Query size

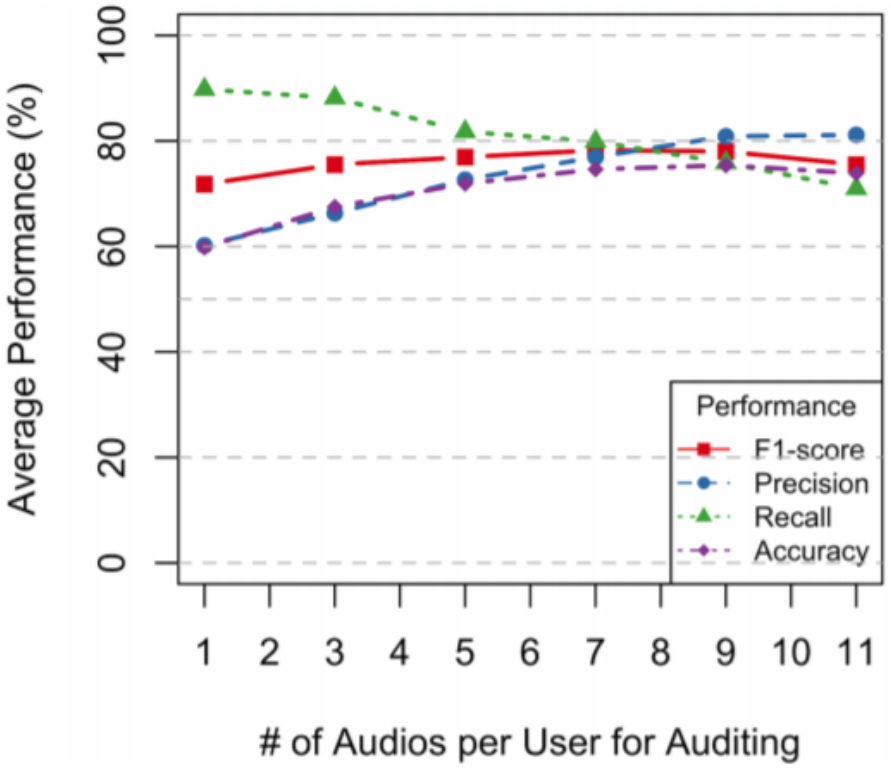


Fig.7. A varying number of audios used for each speaker when querying an auditor model trained with 5 audios per user

Query size

1. More queries results in a better precision
2. Recall and accuracy drops after a certain threshold
    a. Model is “cautious”

1. 9 queries seems to be the optimal spot

RQ3: Transferability

Different datasets distributions from black box

Train Target ASR on different datasets:

- **Librispeech (100 hours clean + 500h noisy)**
  - Recordings of audiobooks
- **TIMIT (6,300 samples)**
  - Telephone conversations
- **TED-LIUM (118 hours)**
  - TED Talks recordings

Auditor is trained on Librispeech

Different ASR training sets

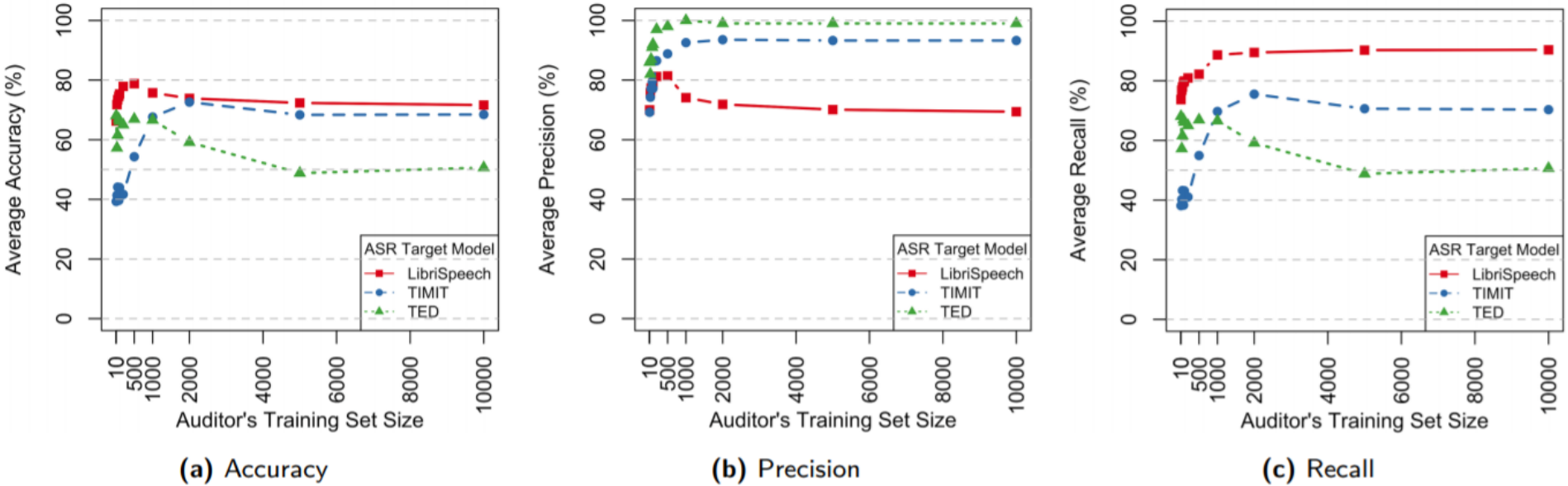


Fig. 5. The auditor model audits target ASR models trained with training sets of different data distributions. We observe that in regards to accuracy and recall the target model with the same distribution as the auditor performs the best, while the contrary is observed for precision. Nevertheless, the data transferability is well observed with reasonably high metrics for all data distributions.

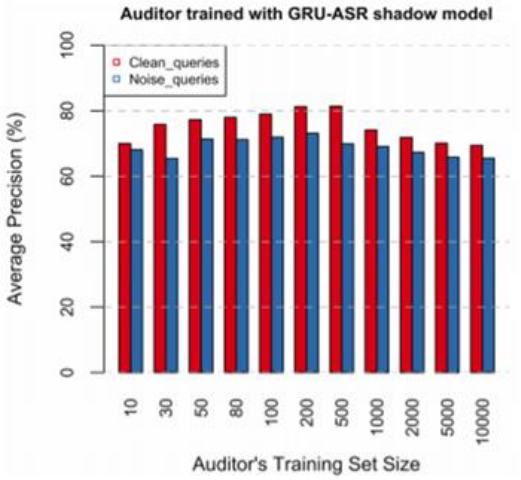
Different ASR training sets

1. Training shadow ASR on the same dataset performs the best
    a. Low precision: tends to classify non members as members

1. Auditor performs well even when trained with different datasets

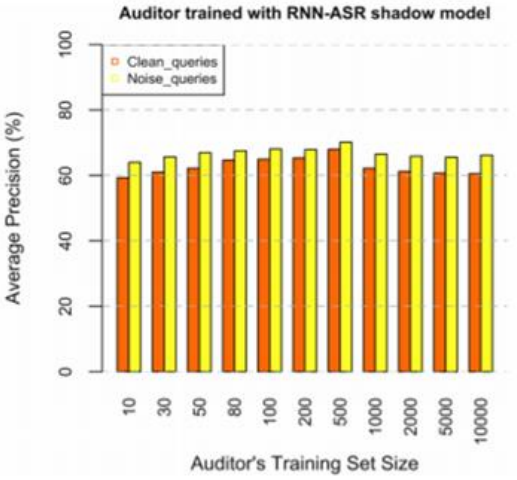
Noisy queries

- ➔ Train two target LSTM-HMM ASR
  - ◆ Trained with clean audios
  - ◆ Trained with noisy audios
- ➔ Train shadows models and create sets for noisy and clean audios
- ➔ Train Shadow Hybrid models with different DNN models
  - ◆ LSTM
  - ◆ GRU
  - ◆ RNN

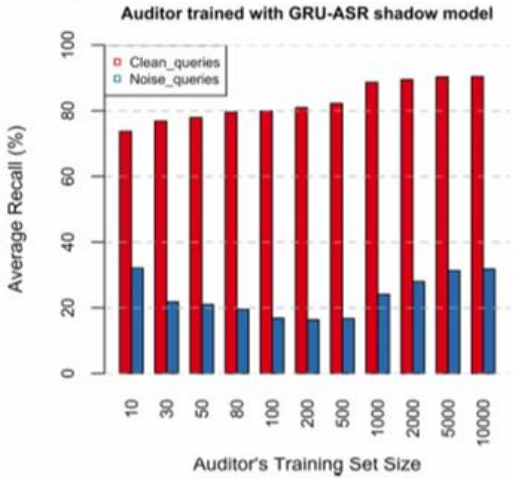


(a) GRU-based Auditor Precision

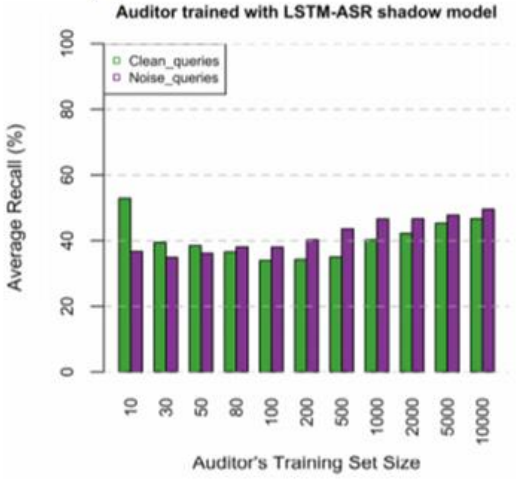
(b) LSTM-based Auditor Precision



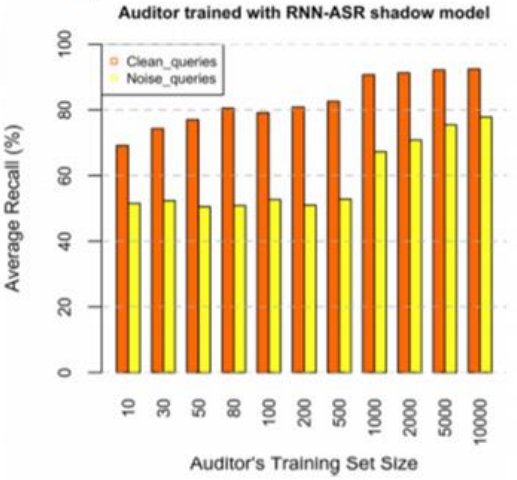
(c) RNN-based Auditor Precision



(d) GRU-based Auditor Recall



(e) LTSM-based Auditor Recall



(f) RNN-based Auditor Recall

Noisy queries

1. Noisy training sets negatively impact the auditor’s recall

1. The auditor model is less impacted by noise when built with the same architecture as target asr

RQ4: Robustness to different blackbox models

Different target ASR architectures

- ➔ Train Shadow Hybrid models with different models
  - LSTM
  - GRU
  - RNN
  - Combine multiple Shadow ASR

Different ASR Shadow architectures

Table 3. Information about ASR models trained with different architectures. (WER_{train}: the prediction's WER on the training set; WER_{test}: the prediction's WER on the testing set; t: target model; s: shadow model.).

ASR Models	Model's Architecture	Dataset Size	WER_{train}	WER_{test}
LSTM-ASR (s)	4-LSTM layer + Softmax	360 hrs	6.48%	9.17%
RNN-ASR (s)	4-RNN layer + Softmax	360 hrs	9.45%	11.09%
GRU-ASR (s)	5-GRU layer + Softmax	360 hrs	5.99%	8.48%
LSTM-ASR (t)	4-LSTM layer + Softmax	100 hrs	5.06%	9.08%

Different ASR Shadow architectures

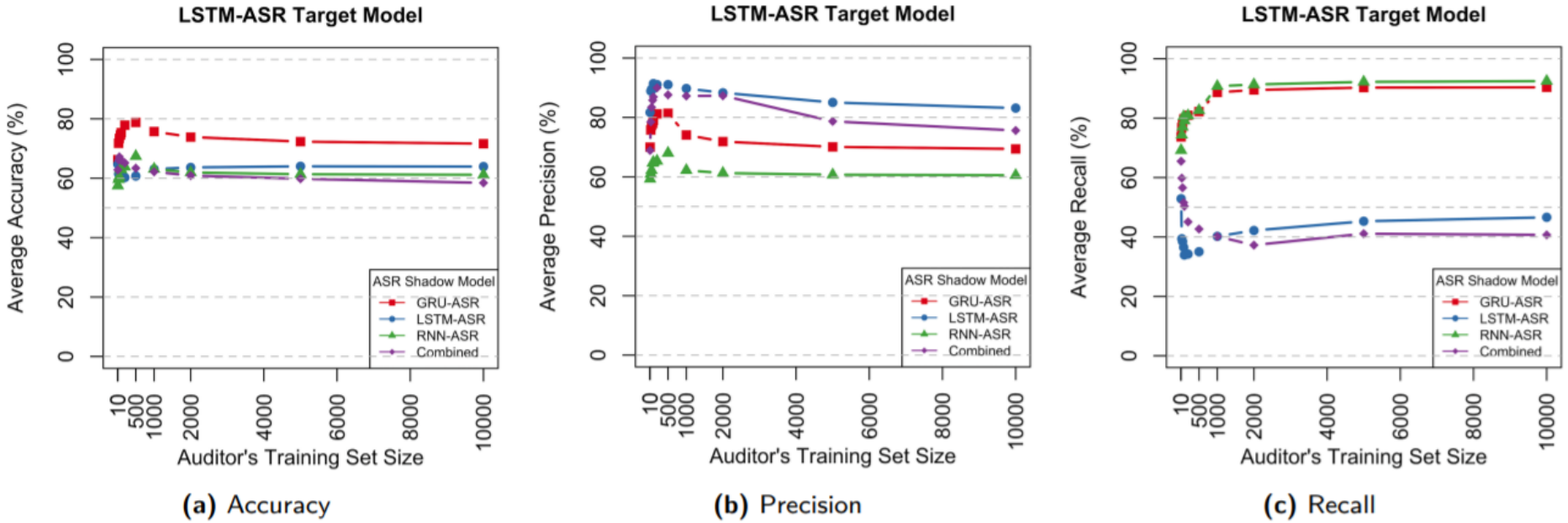


Fig. 8. Different auditor model performance when trained with different ASR shadow model architectures.

Different target ASR pipeline

- ➔ Evaluate Hybrid Shadow ASR and Auditor with different target ASR pipeline
  - ◆ Hybrid DNN-HMM
  - ◆ End-to-End

Different ASR pipelines

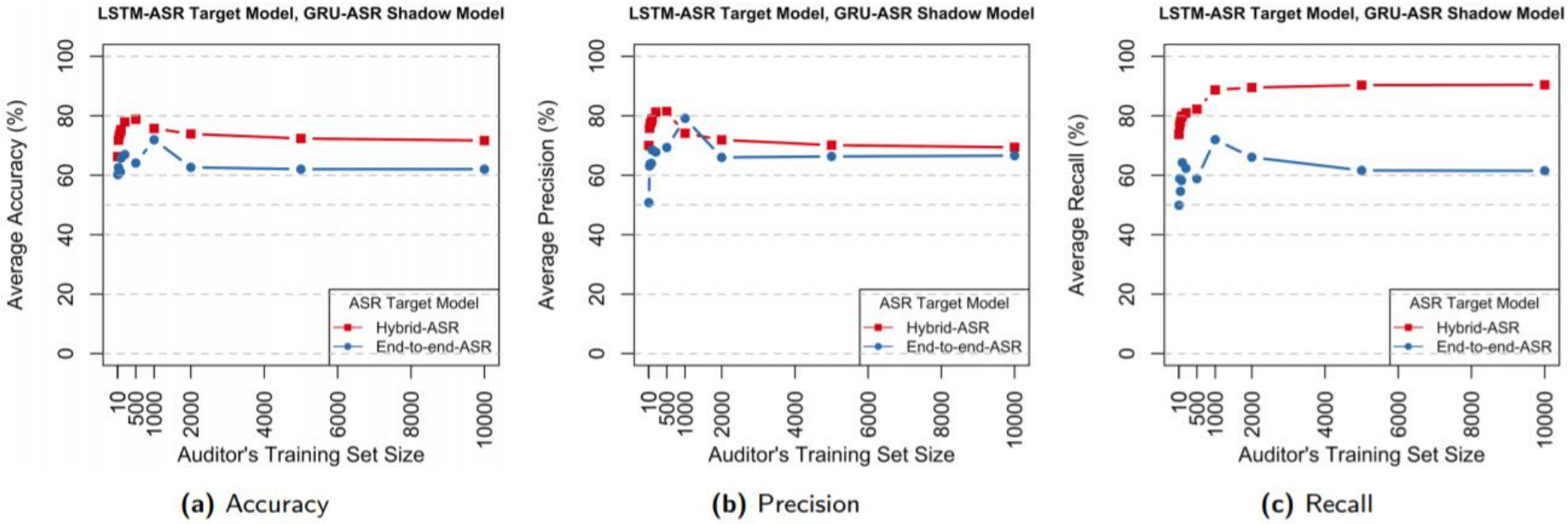


Fig. 10. The audit model audits different target ASR models trained with different pipelines.

Different ASR pipelines

$$\text{Overfitting} = \text{WER}_{\text{train}} - \text{WER}_{\text{test}}$$

Target ASR Model	Overfitting value
Hybrid	0.04
End-to-end	0.14

Robustness to blackbox models

1. Differences in ASR pipelines between the target model and the shadow model negatively impact the performance of the auditor.
2. Different architectures still perform above random
3. Overfitting is not an indication of higher leakage

Real-world evaluation: SIRI

Set up

- ➔ Shadow ASR: GRU shadow trained on Librispeech
- ➔ Sample 5 samples per user with 1000 users
- ➔ Samples played via bluetooth speaker

- ➔ Samples:
  - ◆ Member: Iphone owner
  - ◆ Non-members: 52 speakers from Librispeech
  - ◆ 6 members samples, 52 non member samples

Results

- ➔ Overall **accuracy** is 89.76%
- ➔ Member **precision**: 58.45%
- ➔ Non-member **precision**: 92.61%
- ➔ Precision for seen member samples: 100%

Discussion

1. The auditor performs best on seen audios
2. Training size impact the performance - sweet spot
3. Auditor is relatively robust to different datasets and architectures
4. Noisy training has a negative impact on the auditor’s performance
5. Previous mitigations strategies do not apply here
    a. Overfitting is less indicative of the auditor’s performance

Mitigations

- ➔ Voice conversion
- ➔ Differential Privacy
- ➔ Regularization might not be very effective here

Limitations

1. Performance relies on audio features
    a. More feature exploration needed
2. Some doubts about ground truth for Siri evaluation
3. Limited testing on blackbox models
4. Few mitigations strategies are discussed

Conclusion

1. Work present a user-level membership inference attack on black box voice services
2. Approach it as a ‘tool’ to detect privacy violations
3. Show robustness to different environment, data distribution and model architectures
4. Successfully demonstrate the feasibility of the attack on a real-world bloack box model

Related Work

1. **Sound masking**
    a. Exploiting Sound Masking for Audio Privacy in Smartphones, Tung, Yu-Chih, and Kang G. Shin., Asia CCS '19
    b. Patronus: preventing unauthorized speech recordings with support for selective unscrambling, Li, Lingkun, et al., SenSys '20
2. **Microphone Jamming**:
    a. "Alexa, stop spying on me!": speech privacy protection against voice assistants, Sun, Ke, Chen Chen, and Xinyu Zhang., Sensys 20’

Thank you!