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

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

Abstract: With the rapid development of deep learning techniques, the popularity of voice services implemented on various Internet of Things (IoT) devices is ever increasing. In this paper, we examine user-level membership inference in the problem space of voice services, by designing an audio auditor to verify whether a specific user had unwillingly contributed audio used to train an automatic speech recognition (ASR) model under strict black-box access. With user representation of the input audio data and their corresponding translated text, our trained auditor is effective in user-level audit. We also observe that the auditor trained on specific data can be generalized well regardless of the ASR model architecture. We validate the auditor on ASR models trained with LSTM, RNNs, and GRU algorithms on two state-of-the-art pipelines, the hybrid ASR system and the end-to-end ASR system. Finally, we conduct a real-world trial of our auditor on iPhone Siri, achieving an overall accuracy exceeding 80%. We hope the methodology developed in this paper and findings can inform privacy advocates to overhaul IoT privacy.

Yuantian Miao: Swinburne University of Technology, Australia
Minhui Xue: The University of Adelaide, Australia
***Corresponding Author: Chao Chen:** James Cook University, Australia. Email: chao.chen@jcu.edu.au.
Lei Pan: Deakin University, Australia
Jun Zhang: Swinburne University of Technology, Australia
Benjamin Zi Hao Zhao: The University of New South Wales and CSIRO-Data61, Australia
Dali Kaafar: Macquarie University and CSIRO-Data61, Australia
Yang Xiang: Swinburne University of Technology, Australia

1 Introduction

Automatic speech recognition (ASR) systems are widely adopted on Internet of Things (IoT) devices [16, 20]. In the IoT voice services space, competition in the smart speaker market is heating up between giants like Apple, Microsoft, and Amazon [24]. However parallel to the release of new products, consumers are growing increasingly aware and concerned about their privacy, particularly about unauthorized access to user's audio in these ASR systems. Of late, privacy policies and regulations, such as the General Data Protection Regulations (GDPR) [7], the Children's Online Privacy Protection Act (COPPA) [19], and the California Consumer Privacy Act (CCPA) [3], have been enforced to regulate personal data processing. Specifically, the Right to be Forgotten [48] law allows customers to prevent third-party voice services from continuously using their data [2]. However, the murky boundary between privacy and security can thwart IoT's trustworthiness [13, 34] and many IoT devices may attempt to sniff and analyze the audio captured in real-time without a user's consent [33]. Most recently, on WeChat – a hugely popular messaging platform within China and Worldwide – a scammer camouflaged their voice to sound like an acquaintance by spoofing his or her voice [4]. Additionally, in 2019, The Guardian reported a threat regarding the user recordings leakage via Apple Siri [11]. Auditing if an ASR service provider adheres to its privacy statement can help users to protect their data privacy. It motivates us to develop techniques that enable auditing the use of customers' audio data in ASR models.

Recently, researchers have shown that *record-level membership inference* [10, 32, 39] may expose information about the model's training data even with only black-box access. To mount membership inference attacks, Shokri et al. [39] integrate a plethora of shadow models to constitute the attack model to infer membership, while Salem et al. [32] further relax this process and resort to the target model's confidence scores alone. However, instead of inferring record-level information, we seek to infer user-level information to verify whether a user has any audios within the training set.

Therefore, we define **user-level membership inference** as: *querying with a user's data, if this user has any data within target model's training set, even if the query data are not members of the training set, this user is the user-level member of this training set.*

Song and Shmatikov [40] discuss the application of user-level membership inference on text generative models, exploiting several top ranked outputs of the model. Considering most ASR systems in the real world do not provide the confidence score, significantly differing from text generative models lending confidence scores [40], this paper targets user-level membership inference on ASR systems under *strict black-box access*, which we define as no knowledge about the model, with only knowledge of the model's output *excluding confidence score and rank information*, i.e., only predicted label is known.

Unfortunately, user-level membership inference on ASR systems with strict black-box access is challenging. (i) Lack of information about the target model is challenging [5]. As strict black-box inference has little knowledge about the target model's performance, it is hard for shadow models to mimic a target model. (ii) User-level inference requires a higher level of robustness than record-level inference. Unlike record-level, user-level inference needs to consider the speaker's voice characteristics. (iii) ASR systems are complicated due to their learning architectures [5], causing membership inference with shadow models to be computationally resource and time consuming. Finally, time-series audio data is significantly more complex than textual data, resulting in varied feature patterns [6, 38].

In this paper, we design and evaluate our audio auditor to help users determine whether their audio records have been used to train an ASR model without their consent. We investigate two types of targeted ASR models: a hybrid ASR system and an end-to-end ASR system. With an audio signal input, both of the models transcribe speech into written text. The auditor audits the target model with an intent via strict black-box access to infer user-level membership. The auditor will behave differently depending on whether audio is transcribed from within its training set or from other datasets. Thus, one can analyze the transcriptions and use the outputs to train a binary classifier as the auditor. As our primary focus is to infer user-level membership, instead of using the rank lists of several top output results, we only use one text output, the user's speed, and the input audio's true transcription while analyzing the transcription outputs (see details in Section 3).

In summary, the main contributions of this paper are as follows:

1. We propose the use of user-level membership inference for auditing the ASR model under strict black-box access. With access to the top predicted label only, our audio achieves 78.81% accuracy. In comparison, the best accuracy for the user-level auditor in text generative models with one top-ranked output is 72.3% [40].

2. Our auditor is effective in user-level audit. For the user who has audios within the target model's training set, the accuracy of our auditor querying with these recordings can achieve more than 80%. In addition, only nine queries are needed for each user (regardless of their membership or non-membership) to verify their presence of recordings in the ASR model, at an accuracy of 75.38%.

3. Our strict black-box audit methodology is robust to various architectures and pipelines of the ASR model. We investigate the auditor by auditing the ASR model trained with LSTM, RNNs, and GRU algorithms. In addition, two state-of-the-art pipelines in building ASR models are implemented for validation. The overall accuracy of our auditor achieves approximately 70% across various ASR models on auxiliary and cross-domain datasets.

4. We conduct a proof-of-concept test of our auditor on iPhone Siri, under the strict black-box access, achieving an overall accuracy in excess of 80%. This real-world trial lends evidence to the comprehensive synthetic audit outcomes observed in this paper.

To the best of our knowledge, this is the *first* paper to examine user-level membership inference in the problem space of voice services. We hope the methodology developed in this paper and findings can inform privacy advocates to overhaul IoT privacy.

2 Background

In this section, we overview the automatic speech recognition models and membership inference attacks.

2.1 The Automatic Speech Recognition Model

There are two state-of-the-art pipelines used to build the automatic speech recognition (ASR) system, including the typical hybrid ASR systems and end-to-end ASR systems [27]. To test the robustness of our auditor,
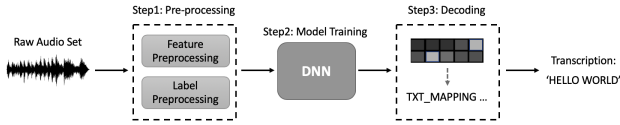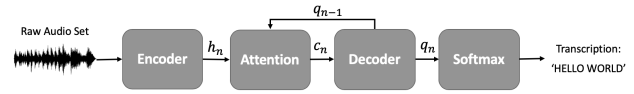
(a) A hybrid ASR system. There are three main steps: (i) the preprocessing step extracts features to represent the raw audio data, (ii) the DNN training step trains the acoustic model and calculates the pseudo-posteriors, and (iii) the decoding step aims to map the predicted symbol combinations to texts and output the transcription results with the highest score.

(b) An end-to-end system. There are four main components: (i) the encoder transforms the input audio into a high-level representation h_n, (ii) the attention mechanism integrates the representation h_n with previous decoder output q_{n-1} and gain the context output c_n, (iii) the decoder decodes the context output c_n with previous ground truth as q_n, and (iv) the Softmax activation like CharDistribution predicts q_n auto-regressively and results in final transcription.

Fig. 1. Two state-of-the-art ASR systems

we implement both open-source hybrid and end-to-end ASR systems focusing on a speech-to-text task as the target models.

Hybrid ASR Systems are mainly DNN-HMM-based acoustic models [46]. As shown in Fig. 1a, typically, a hybrid ASR system is composed of a preprocessing step, a model training step, and a decoding step [35]. During the preprocessing step, features are extracted from the input audio, while the corresponding text is processed as the audio's label. The model training step trains a DNN model to create HMM class posterior probabilities. The decoding step maps these HMM state probabilities to a text sequence. In this work, the hybrid ASR system is built using the pytorch-kaldi speech recognition toolkit [29]. Specifically, feature extraction transforms the audio frame into the frequency domain, as Mel-Frequency Cepstral Coefficients (MFCCs) features. For an additional processing step, feature-space Maximum Likelihood Linear Regression (fMLLR) is used for speaker adaptation. Three popular neural network algorithms are used to build the acoustic model, including Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNNs). The decoder involves a language model which provides a language probability to re-evaluate the acoustic score. The final transcription output is the sequence of the most suited language with the highest score.

End-to-End ASR Systems are attention-based encoder-decoder models [14]. Unlike hybrid ASR systems, the end-to-end system predicts sub-word sequences which are converted directly as word sequences. As shown in Fig. 1b, the end-to-end system is a unified neural network modeling framework containing four components: an encoder, an attention mechanism, a decoder, and a Softmax layer. The encoder contains feature extraction (i.e., VGG extractor) and a few neural network layers (i.e., BiLSTM layers), which encode the input audio into high-level representations. The location-aware attention mechanism integrates the rep-

resentation of this time frame with the previous decoder outputs. Then the attention mechanism can output the context vector. The decoder can be a single layer neural network (i.e., an LSTM layer), decoding the current context output with the ground truth of last time frame. Finally, the softmax activation, which can be considered as "CharDistribution", predicts several outputs and integrates them into a single sequence as the final transcription.

2.2 Membership Inference Attack

The membership inference attack is considered as a significant privacy threat for machine learning (ML) models [22]. The attack aims to determine whether a specific data sample is within the target model's training set or not. The attack is driven by the different behaviors of the target model when making predictions on samples within or out of its training set.

Various membership inference attack methods have been recently proposed. Shokri et al. [39] train shadow models to constitute the attack model against a target ML model with black-box access. The shadow models mimic the target model's prediction behavior. To improve accuracy, Liu et al. [15] and Hayes et al. [10] leverage Generative Adversarial Networks (GAN) to generate shadow models with increasingly similar outputs to the target model. Salem et al. [32] relax the attack assumptions mentioned in the work [39], demonstrating that shadow models are not necessary to launch the membership inference attack. Instead, a threshold of the predicted confidence score can be defined to substitute the attack model. Intuitively, a large confidence score indicates the sample as a member of the training set [41]. The attacks mentioned in the work above are all performed on the record level, while Song and Shmatikov [40] study a user-level membership inference attack against text generative models. Instead of using the prediction label along with the confidence score,

Song and Shmatikov [40] utilize word's rank list information of several top-ranked predictions as key features to generate the shadow model. Apart from the black-box access, Farokhi and Kaafar [8] model the record-level membership inference attack under the white-box access.

Unlike image recognition systems or text generative systems, ASR systems present additional challenges [5]. With strict black-box access, attacks using confidence scores cannot be applied. With limited discriminative power, features can only be extracted from the predicted transcription and its input audio to launch membership inference attacks, i.e., audio auditing in our paper.

3 Auditing the ASR Models

In this section, we first formalize our objective for auditing ASR models. Secondly, we present how a user-level ASR auditor can be constructed and used to audit the target ASR. Finally, we show how we implement the auditor.

3.1 Problem Statement

We define *user-level membership inference* as querying a user's data and trying to determine whether any data within the target model's training set belongs to this user. Even if the queried data is not members of the training set, but data belonging to this user is members in the training set, then this user is regarded as the user-level member of this training set. Let $(x, y) \in \mathbb{X} \times \mathbb{Y}$ denote an audio sample, where x presents the audio component, and y is the actual text of x. Assume an ASR model is a function $\mathcal{F} : \mathbb{X} \to \mathbb{Y}$. $\mathcal{F}(x)$ is the model's translated text. The smaller the difference between $\mathcal{F}(x)$ and y, the better the ASR model performs. Let D represent a distribution of audio samples. Assume an audio set A is sampled from D of size N $(A \sim D^N)$. Let U be the speaker set of A of size M $(U \leftarrow A)$. The ASR model trained with the dataset A is denoted as $\mathcal{F}_A$. Let $\mathcal{A}$ represent our auditor, and the user-level auditing process can be formalized as:

- A speaker u has $S = \bigcup_{i=1}^{m}(x_i, y_i)$, where $u \leftarrow S$.
- Let $Y' = \bigcup_{i=1}^{m} y_i'$, when $y_i' = \mathcal{F}_A(x_i)$.
- Let "member" $= 0$ and "nonmember" $= 1$.
- Set $r = 0$ if $u \in U$, or $r = 1$ if $u \notin U$.

- The auditor successes if $\mathcal{A}(u, S, Y') = r$; otherwise it fails.

Our auditor, as an application of user-level membership inference, checks a speaker's membership of an ASR model's training set. This ASR model is considered as the target model. To closely mirror the real world, we query the target model with strict black-box access. The model only outputs a possible text sequence as its transcription when submitting an audio sample to the target model. This setting reflects the reality, as the auditor may not know this transcription's posterior probabilities or other possible transcriptions. Additionally, any information about the target model is unknown, including the model's parameters, algorithms used to build the model, and the model's architecture. To evaluate our auditor, we develop our target ASR model $\mathcal{F}_{tar}$ using an audio set A_{tar} with two popular pipelines — hybrid ASR model and end-to-end ASR model — to represent the ASR model in the real world. As described in Section 2, the hybrid ASR model and the end-to-end ASR model translate the audio in different manners. Under the strict black-box access, the auditor only knows query audio records of a particular user u and its corresponding output transcription. The goal of the auditor is to build a binary classifier $\mathcal{A}_{audit}$ to discriminate whether this user is the member of the user set in which their audio records have been used as target model's training data $(u \in U_{tar}, U_{tar} \leftarrow A_{tar})$.

3.2 Overview of the Proposed Audio Auditor

The nature of membership inference [39] is to learn the difference of a model fed with its actual training samples and other samples. User-level membership inference, like its record-level variant, requires higher robustness. Apart from the disparity of the target model's performance on record-level, our auditor needs to consider the speaker's characteristics as well. Since the posterior probabilities (or confidence scores) are not part of the outputs, shadow models are necessary to audit the ASR model.

Fig. 2 depicts a workflow of our audio auditor auditing an ASR model. Generally, there are two processes, i.e., training and auditing. The former process is to build a binary classifier as a user-level membership auditor $\mathcal{A}_{audit}$ using a supervised learning algorithm. The latter uses this auditor to audit an ASR model $\mathcal{F}_{tar}$ by querying a few audios spoken by one user u. In Sec-
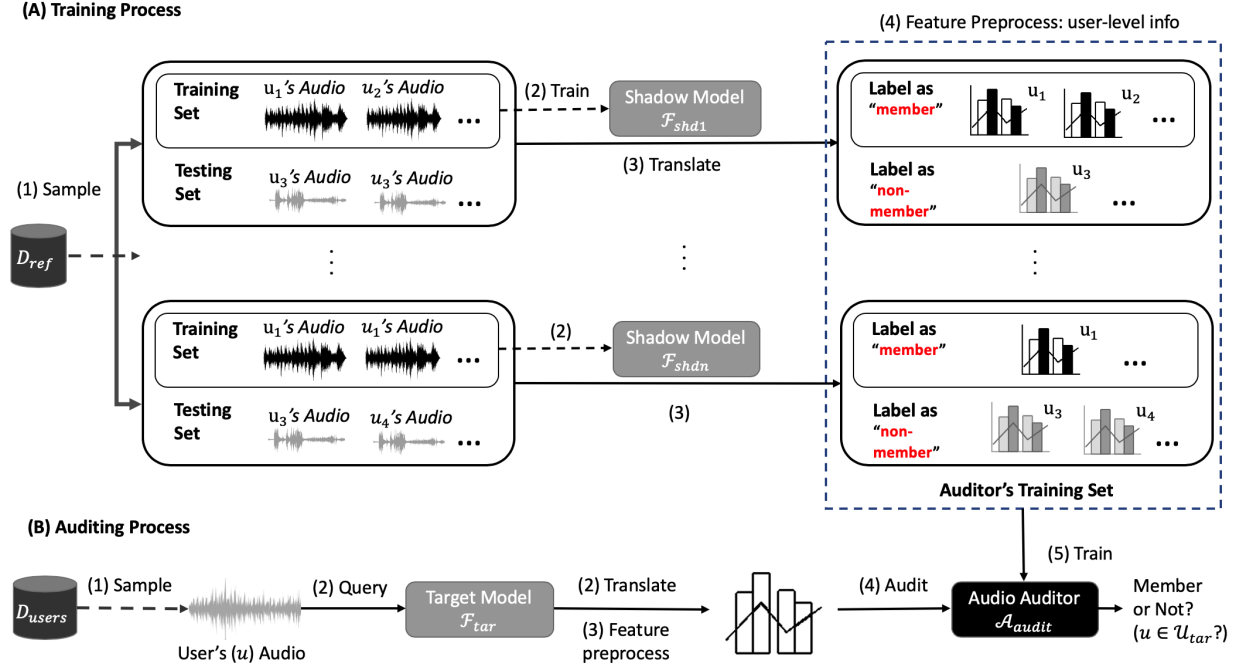
(A) Training Process

Fig. 2. Auditing an ASR model. (i) In the training process, we sample n datasets from the auxiliary reference dataset D_{ref} to build n ($n \geqslant 1$) shadow models. Each shadow model dataset $A_{shdi} \sim D_{ref}^N, i = 1, \ldots, n$ is split to a training set A_{shdi}^{train} and a testing set A_{shdi}^{test}. Then we query the shadow model with A_{shdi}^{test} and A_{shdi}^{train} and label their transcriptions as "member" or "nonmember". Then an audit model can be trained with the outputs of shadow models. (ii) In the auditing process, we randomly sample a particular speaker's (u's) audios $A_u \sim D_{users}$ to query our target ASR model. Feature vectors from outputs of the target ASR model can be passed to the audit model to determine whether $u \in U_{tar} \leftarrow A_{tar}$ holds.

tion 4.4, we show that only a small number of audios per user can determine whether $u \in U_{tar}$ or $u \notin U_{tar}$. Furthermore, a small number of users used to train the auditor is sufficient to provide a satisfying result.

Training Process. The primary task in the training process is to build up shadow models of high quality. Shadow models, mimicking the target model's behaviors, try to infer the targeted ASR model's decision boundary. Due to strict black-box access, a good quality shadow model performs with an approximate testing accuracy as the target model. We randomly sample n datasets from the auxiliary reference dataset D_{ref} as $A_{shd1}, \ldots, A_{shdn}$ to build n shadow models. Each shadow model's audio dataset $A_{shdi}, i = 1, \ldots, n$ is split to a training set A_{shdi}^{train} and a testing set A_{shdi}^{test}. To build up the ground truth for auditing, we query the shadow model with A_{shdi}^{train} and A_{shdi}^{test}. Assume a user's audio set A_u is sampled from users' audio sets D_{users}. According to the user-level membership inference definition, the outputs from the audio $A_u \in A_{shdi}^{test}$ where its speaker $u \notin U_{shdi}^{train}$ are labeled as "nonmember". Otherwise, the outputs translated from the audio $A_u \in A_{shdi}^{train}$ and from the audio $A_u \in A_{shdi}^{test}$ where its speaker $u \in U_{shdi}^{train}$ are all labeled as "member". Herein, $U_{shdi}^{train} \leftarrow$

A_{shdi}^{train}. To simplify the experiment, for each shadow model, training samples are disjoint from testing samples ($A_{shdi}^{train} \cap A_{shdi}^{test} = \emptyset$). Their user sets are disjoint as well ($U_{shdi}^{train} \cap U_{shdi}^{test} = \emptyset$). With some feature extraction (noted below), those labeled records are gathered as the auditor model's training set.

Feature extraction is another essential task in the training process. Under the strict black-box access, features are extracted from the input audio, ground truth transcription, and the predicted transcription. As a user-level membership inferrer, our auditor needs to learn the information about the target model's performance and the speaker's characteristics. Comparing the ground truth transcription and the output transcription, the **similarity score** is the first feature to represent the ASR model's performance. To compute the two transcriptions' similarity score, the GloVe model [26] is used to learn the vector space representation of these two transcriptions. Then the cosine similarity distance is calculated as the two transcriptions' similarity score. Additionally, the input **audio frame length** and the **speaking speed** are selected as two features to present the speaker's characteristics. Because a user almost always provides several audios to train the ASR model,

Table 1. The audit model's performance when selecting either 3 features, 5 features, or 5 features with MFCCs for each audio's query.

	F1-score	Precision	Recall	Accuracy
Feature_Set3	63.89%	68.48%	60.84%	61.13%
Feature_Set5	81.66%	81.40%	82.22%	78.81%
Feature_Set5 + MFCCs	81.01%	79.72%	82.52%	77.82%

Table 2. The audit model's performance trained with different algorithms.

	F1-score	Precision	Recall	Accuracy
DT	68.67%	70.62%	67.29%	64.97%
RF	81.66%	81.40%	82.22%	78.81%
3-NN	58.62%	64.49%	54.69%	56.16%
NB	34.42%	93.55%	21.09%	53.96%

statistical calculation is applied to the three features above, including sum, maximum, minimum, average, median, standard deviation, and variance. After the feature extraction, all user-level records are gathered with labels to train an auditor model using a supervised learning algorithm.

To test the quality of the feature set above, we trained an auditor with 500 user-level samples using the Random Forest (RF) algorithm. By randomly selecting 500 samples 100 times, we achieve an average accuracy result over 60%. Apart from the three aforementioned features, two additional features are added to capture more variations in the model's performance, including **missing characters** and **extra characters** obtained from the transcriptions. For example, if *(truth transcription, predicted transcription) = (THAT IS KAFFAR'S KNIFE, THAT IS CALF OUR'S KNIFE)*, then *(missing characters, extra characters) = (KFA, CL OU)*. Herein, the blank character in the extra characters means that one word was mistranslated as two words. With these two extra features, a total of five features are extracted from record-level samples: *similarity score, missing characters, extra characters, frame length,* and *speed*. The record-level samples are transformed into user-level samples using statistical calculation as previously described. We compare the performance of two auditors trained with the two feature sets. We also consider adding 13 Mel-Frequency Cepstral Coefficients (MFCCs) as the additional audio-specific feature set to accentuate each user's records with the average statistics. As seen in Table 1, the statistical feature set with 5-tuple is the best choice with approximately 80% accuracy, while the results with additional audio-specific features are similar, but trail by one percentage. Thus, we proceed with five statistical features to represent each user as the outcome of the feature extraction step.

Auditing Process. After training an auditor model, we randomly sample a particular speaker's ($u's$) audios A_u from D_{users} to query our target ASR model. With the same feature extraction, the outputs can be passed to the auditor model to determine whether this speaker $u \in U_{tar}$. We assume that our target model's dataset D_{tar} is disjoint from the auxiliary reference dataset D_{ref} ($D_{tar} \cap D_{ref} = \emptyset$). In addition, U_{ref} and U_{tar} are also disjoint ($U_{tar} \cap U_{ref} = \emptyset$). For each user, as we will show, only a limited number of audios are needed to query the target model and complete the whole auditing phase.

3.3 Implementation

In these experiments, we take audios from LibriSpeech [25], TIMIT [9], and TED-LIUM [30] to build our target ASR model and the shadow model. Detailed information about the speech corpora and model architectures can be found in the Appendix. Since the LibriSpeech corpus has the largest audio sets, we primarily source records from LibriSpeech to build our shadow models.

Target Model. Our target model is a speech-to-text ASR model. The inputs are a set of audio files with their corresponding transcriptions as labels, while the outputs are the transcribed sequential texts. To simulate most of the current ASR models in the real world, we created a state-of-the-art hybrid ASR model [35] using the PyTorch-Kaldi Speech Recognition Toolkit [29] and an end-to-end ASR model using the Pytorch implementation [14]. In the preprocessing step, fMLLR features were used to train the ASR model with 24 training epochs. Then, we trained an ASR model using a deep neural network with four hidden layers and one Softmax layer. We experimentally tuned the batch size, learning rate and optimization function to gain a model with better ASR performance. To mimic the ASR model in the wild, we tuned the parameters until the training accuracy exceeded 80%, similar to the results shown in [14, 27]. Additionally, to better contextualize our audit results, we report the overfitting level of the ASR models, defined as the difference between the predictions' Word Error Rate (WER) on the training set and the testing set ($Overfitting = WER_{train} - WER_{test}$).

4 Experimental Evaluation and Results

The goal of this work is to develop an auditor for users to inspect whether their audio information is used without consent by ASR models or not. We mainly focus on the evaluation of the auditor, especially in terms of its effectiveness, efficiency, and robustness. As such, we pose the following research questions.

- *The effectiveness of the auditor.* We train our auditor using different ML algorithms and select one with the best performance. How does the auditor perform with different sizes of training sets? How does it perform in the real-world scenario, such as auditing iPhone Siri?
- *The efficiency of the auditor.* How many pieces of audios does a user need for querying the ASR model and the auditor to gain a satisfying result?
- *The data transferability of the auditor.* If the data distribution of the target ASR model's training set is different from that of the auditor, is there any effect on the auditor's performance? If there is a negative effect on the auditor, is there any approach to mitigate it?
- *The robustness of the auditor.* How does the auditor perform when auditing the ASR model built with different architectures and pipelines? How does an an auditor perform when a user queries the auditor with audios recorded in a noisy environment (i.e., noisy queries)?

4.1 Effect of the ML Algorithm Choice for the Auditor

We evaluate our audio auditor as a user-level membership inference model against the target ASR system. This inference model is posed as a binary classification problem, which can be trained with a supervised ML algorithm. We first consider the effect of different training algorithms on our auditor performance.

To test the effect of different algorithms on our audit methodology, we need to train one shadow ASR model for training the auditor and one target ASR model for the auditor's auditing phase. We assume the target ASR model is a hybrid ASR system whose acoustic model is trained with a four-layer LSTM network. The training set used for the target ASR model is 100 hours of clean audio sampled from the LibriSpeech corpus [25].

Additionally, the shadow model is trained using a hybrid ASR structure where GRU network is used to build its acoustic model. According to our audit methodology demonstrated in Fig. 2, we observe the various performance of the audio auditor trained with four popular supervised ML algorithms listing as Decision Tree (DT), Random Forest (RF), k-Nearest Neighbor where $k = 3$ (3-NN), and Naive Bayes (NB). After feature extraction, 500 users' samples from the shadow model's query results are randomly selected as the auditor's training set. To avoid potential bias in the auditor, the number of "member" samples and the number of "nonmember" samples are equal in all training set splits ($\#\{u \in U_{shd}\} = \#\{u \notin U_{shd}\}$). An additional step taken to eliminate bias, is that each experimental configuration is repeated 100 times. Their average result is reported as the respective auditor's final performance in Table 2.

As shown in Table 2, our four metrics of accuracy, precision, recall, and F1-score are used to evaluate the audio auditor. In general, the RF auditor achieves the best performance compared to the other algorithms. Specifically, the accuracy approaches 80%, with the other three metrics also exceeding 80%. We note that all auditors' accuracy results exceed the random guess (50%). Aside from the RF and DT auditors, the auditor with other ML algorithms behaves significantly differently in terms of precision and recall, where the gaps of the two metrics are above 10%. The reason is in part due to the difficulty in distinguishing the "member" and "nonmember" as a user's audios are all transcribed well at a low speed with short sentences. Tree-based algorithms, with right sequences of conditions, may be more suitable to discriminate the membership. We regard the RF construction of the auditor as being the most successful; as such, RF is the chosen audio auditor algorithm for the remaining experiments.

4.2 Effect of the Number of Users Used in Training Set of the Auditor

To study the effect of the number of users, we assume that our target model and shadow model are trained using the same architecture (hybrid ASR system). However, due to the strict black-box access to the target model, the shadow model and acoustic model shall be trained using different networks. Specifically, LSTM networks are used to train the acoustic model of the target ASR system, while a GRU network is used for the shadow model. As depicted in Fig. 3, each training
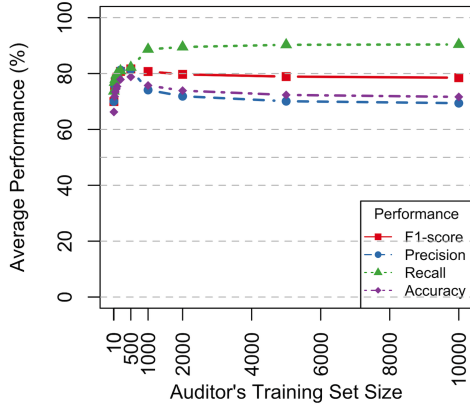
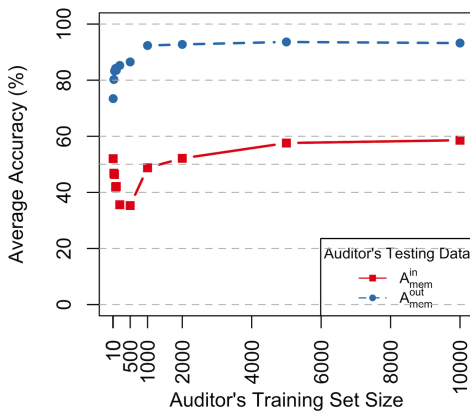**Fig. 3.** Auditor model performance with varied training set size.

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

sample (x_i, y_i) is formed by calculating shadow model's querying results for each user $u_j \leftarrow \bigcup_{i=1}^{m}(x_i, y_i)$. Herein, we train the audio auditor with a varying number of users $(j = 1, ..., M)$. The amount of users M we considered in the auditor training set is 10, 30, 50, 80, 100, 200, 500, 1,000, 2,000, 5,000, and 10,000.

On the smaller numbers of users in the auditor's training set, from Fig. 3, we observe a rapid increase in performance with an increasing number of users. Herein, the average accuracy of the auditor is 66.24% initially, reaching 78.81% when the training set size is 500 users. From 500 users, the accuracy decreases then plateaus. Overall, the accuracy is better than the random guess baseline of 50% for all algorithms. Aside from accuracy, the precision increases from 69.99% to 80.40%; the recall is 73.69% initially and eventually approaches approximately 90%; and the F1-score is about 80% when the training set size exceeds 200. In summary, *we identify the auditor's peak performance when using a relatively small number of users for training.*

Recall the definition of user-level membership in Section 3.1. We further consider two extreme scenarios in the auditor's testing set. One extreme case is that the auditor's testing set only contains member users querying with the unseen audios (excluded from the target model's training set), henceforth denoted as A_{mem}^{out}. The other extreme case is an auditor's testing set that only contains member users querying with the seen audios (exclusively from the target model's training set), herein marked as A_{mem}^{in}. Fig. 4 reports the accuracy of our auditor on A_{mem}^{in} versus A_{mem}^{out}. If an ASR model were to use a user's recordings as its training samples (A_{mem}^{in}), the auditor can determine the user-level membership with a much higher accuracy, when compared to user queries on the ASR model with A_{mem}^{out}. Specifically, A_{mem}^{in} has a peak accuracy of 93.62% when the auditor's training set size M is 5,000. Considering the peak performance previously shown in Fig. 3, auditing with A_{mem}^{in} still achieves good accuracy (around 85%) despite a relatively small training set size. Comparing the results shown in Fig. 3 and Fig. 4, we can infer that the larger the auditor's training set size is, the more likely nonmember users are to be misclassified. The auditor's overall performance peak when using a small number of the training set is largely due to the high accuracy of the shadow model. A large number of training samples perhaps contain the large proportion of nonmember users' records whose translation accuracy is similar to the member users'.

> Overall, it is better for users to choose audios that have a higher likelihood of being contained within the ASR model for audit (for example, the audios once heard by the model).

4.3 Effect of the Target Model Trained with Different Data Distributions

The previous experiment draws conclusions based on the assumption that the distributions of training sets for the shadow model and the target model are the same. That is, these two sets were sampled from LibriSpeech corpus $D_{\mathscr{L}}$ ($A_{tar} \sim D_{\mathscr{L}}$, $A_{shd} \sim D_{\mathscr{L}}$, $A_{tar} \cap A_{shd} = \emptyset$). Aside from the effects that a changing number of users used to train the auditor, we relax this distribution assumption to evaluate the data transferability of the auditor. To this end, we train one auditor using a training set sampled from LibriSpeech $D_{\mathscr{L}}$ ($A_{shd} \sim D_{\mathscr{L}}$). Three

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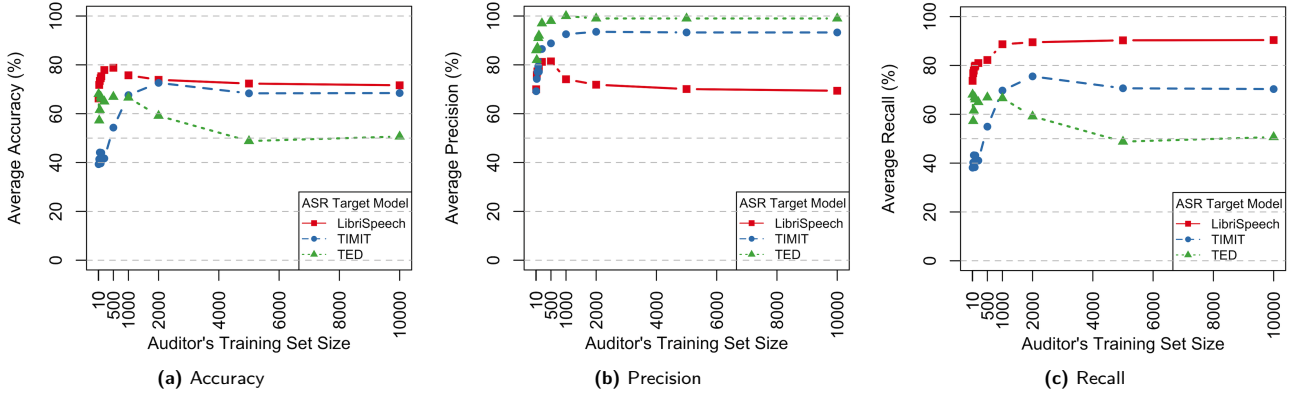
Fig. 5. The auditor model audits target ASR models trained with training sets of different data distributions. We observe that in regards to accuracy and recall the target model with the same distribution as the auditor performs the best, while the contrary is observed for precision. Nevertheless, the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

different target ASR models are built using data selected from LibriSpeech, TIMIT, and TED, respectively.

Fig. 5 plots the auditor's data transferability on average accuracy, precision, and recall. Once above a certain threshold of the training set size (≈ 10), the performance of our auditor significantly improves with an increasing number of users data selected as its user-level training samples. Comparing the peak results, the audit of the target model trained with the same data distribution (LibriSpeech) slightly outperforms the audit of target models with different distributions (TIMIT and TED). For instance, the average accuracy of the auditor auditing LibriSpeech data reaches 78.81% when training set size is 500, while the average audit accuracy of the TIMIT target model peaks at 72.62% for 2,000 users. Lastly the average audit accuracy of TED target model reaches its maximum of 66.92% with 500 users. As shown in Fig. 5, the peaks of precision of the LibriSpeech, TIMIT, and TED target model are 81.40%, 93.54%, and 100%, respectively, opposite of what was observed with accuracy and recall. The observation of the TED target model with extremely high precision and low recall is perhaps due to the dataset's characteristics, where all of the audio clips of TED are long speeches recorded in a noisy environment.

> In conclusion, our auditor demonstrates satisfying data transferability in general.
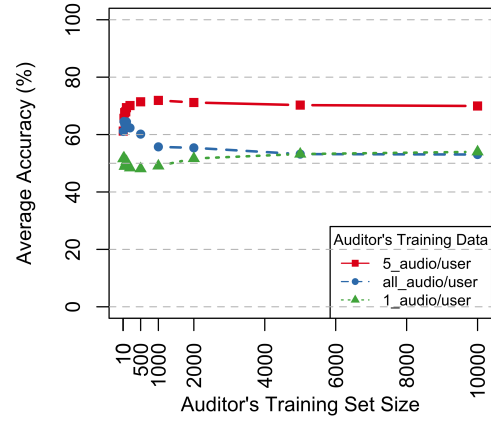
Fig. 6. A comparison of average accuracy for one audio, five audios, and all audios per user when training the auditor model with a limited number of audios per user gained in the auditing phase.
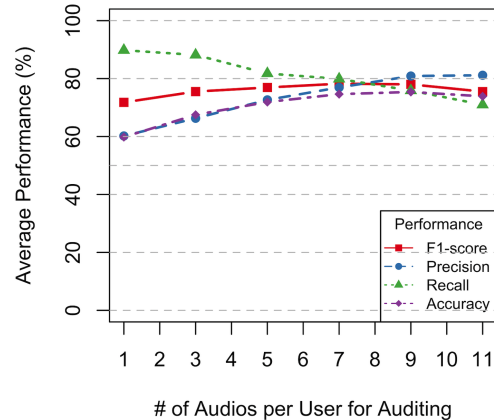
Fig. 7. A varying number of audios used for each speaker when querying an auditor model trained with 5 audios per user.

4.4 Effect of the Number of Audio Records per User

The fewer audio samples a speaker is required to submit for their user-level query during the auditing phase, the more convenient it is for users to use the auditor. Additionally, if the auditor can be trained with user-level training samples accumulated from a reduced number of audios per user, both added convenience and the efficiency of feature preprocessing during the auditor's training can be realized.

A limited number of audio samples per user versus a large number of audio samples per user. Assuming that each user audits their target ASR model by querying with a limited number of audios, we shall consider whether a small number or a large number of audio samples per user should be collected to train our auditor. Herein, varying the number of audios per user only affects the user-level information learned by the auditor during the training phase. To evaluate this, we have sampled one, five, and all audios per user as the training sets when the querying set uses five audios per user. Fig. 6 compares the average accuracy of the auditors when their training sets are processed from limits of one audio, five audios, and finally all audios of each user. To set up the five audio auditor's training sets, we randomly select five audios recorded from each user $u_j \leftarrow \bigcup_{i=1}^{m=5}(x_i, y_i)$, then translate these audios using the shadow model to produce five transcriptions. Following the feature preprocessing demonstrated in Section 3, user-level information for each user is extracted from these five output transcriptions with their corresponding input audios. The same process is applied to construct the auditor in which the training data consists of one audio per user. To set up the auditor's training set with all the users' samples, we collect all audios spoken by each user and repeat the process mentioned above (average $\bar{m} > 62$). Moreover, since the two auditors' settings above rely on randomly selected users, each configuration is repeated 100 times, with users sampled anew, to report the average result free of sampling biases.

Fig. 6 demonstrates that the auditor performs best when leveraging five audios per user during the feature preprocessing stage. When a small number of users are present in the training set, the performance of the two auditors is fairly similar, except the auditor trained with one audio per user. For example, when only ten users are randomly selected to train the auditor, the average accuracy of these two auditors are 61.21% and 61.11%. When increasing to 30 users in the training set, the average accuracy of the 5-sample and all-sample auditors

is 65.65% and 64.56%, respectively. However, with more than 30 users in the training set, the auditor trained on five audios per user outperforms that using all audios per user. Specifically, when using five audios per user, the auditor's average accuracy rises to ≈70% with a larger training set size; compared to the auditor using all audios per user, with a degraded accuracy of ≈55%. This is in part owing to the difficulty of accurately characterizing users' audios. In conclusion, despite restrictions on the number of user audio samples when training the auditor, the auditor can achieve superior performance. Consequently, we recommend that the number of audios per user collected for the auditor's training process should be the same for the auditor's querying process.

A limited number of audio samples per user while querying the auditor. While we have investigated the effect of using a limited number of audios per user to build the training set, we now ask how the auditor performs with a reduced number of audios provided by the user during the querying stage of the audit process, and how many audios each user needs to submit to preserve the performance of this auditor. We assume our auditor has been trained using the training set computed with five audios per user. Fig. 7 displays performance trends (accuracy, precision, recall, and F1-score) when a varying number of query audios per user is provided to the target model. We randomly select a user's audios to query the target model by testing $m = 1, 3, 5, 7, 9$, or 11 audios per user. As Fig. 6 reveals that the accuracy results are stable when training set size is large, we conduct our experiments using 10,000 records in the auditor training set. Again, each experiment is repeated 100 times, and the results are all averaged.

Fig. 7 illustrates that the auditor performs well with results all above 60%. Apart from recall, the other three performances trend upwards with an increasing number of audios per user. The scores of the accuracy, precision, and F1-score are approximately 75%, 81%, and 78%, respectively, when each user queries the target model with nine audios, indicating an improvement over the accuracy (≈72%) we previously observed in Fig. 3. It appears, for accuracy, when the number of query audios per user grows, the upward trend slows down and even slightly declines. The recall is maximized (89.78%) with only one audio queried by each user, decreasing to 70.97% with eleven audios queried for each user. It might happen because the increased number of audio per user does not mean the increased number of users (i.e., testing samples). Since the auditor was trained with five audios per user, the auditor may fail to recog-

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%

nize the user's membership when querying with many more audios.

> Overall, with only a limited number of audios used for audit, e.g. nine audios per user, our auditor still effectively discriminates a user's membership in the target model's training set.

4.5 Effect of Training Shadow Models across Different Architectures

A shadow model trained with different architectures influences how well it mimics the target model and the performance of the user-level audio auditor. In this subsection, we experiment with different shadow model architectures by training the auditor with information from various network algorithms, like LSTM, RNNs, GRU. If the choice of a shadow model algorithm has a substantial impact on the auditor's performance, we shall seek a method to lessen such an impact. We also seek to evaluate the influence of the combining attack as proposed by Salem et al. [32], by combining the transcription results from a set of ASR shadow models, instead of one, to construct the auditor's training set. The feature extraction method is demonstrated in Section 3. We refer to this combination as *user-level combining audit.*

To explore the specific impact of architecture, we assume that the acoustic model of the target ASR system is mainly built with the LSTM network (we call this model the LSTM-ASR Target model). We consider three popular algorithms, LSTM, RNNs, and GRU networks, to be prepared for the shadow model's acoustic model. The details of the target and shadow ASR models above are displayed in Table 3. Each shadow model is used to translate various audios, with their results processed into the user-level information to train an auditor. Consider this, the shadow model that mainly uses the GRU network structure to train its acoustic model is

marked as the GRU-ASR shadow model; its corresponding auditor, named GRU-based auditor, is built using the training set constructed from GRU-ASR shadow model's query results. Our other two auditors have a similar naming convention, an LSTM-based auditor and an RNN-based auditor. Moreover, as demonstrated in Fig. 2, we combine these three shadow models' results ($n = 3$), and construct user-level training samples to train a new combined auditor. This auditor is denoted as the *Combined Auditor* that learns all kinds of popular ASR models.

Fig. 8 demonstrates the varied auditor performance (accuracy, precision and recall) when shadow models using various algorithms are deployed. For accuracy, all four auditors show an upward trend with a small training set size. The peak is observed at 500 training samples then decays to a stable smaller value at huge training set sizes. The GRU-based auditor surpasses the other three auditors in terms of accuracy, with the Combined Auditor performing the second-best when the auditor's training set size is smaller than 500. As for precision, all experiments show relatively high values (all above 60%), particularly the LSTM-based auditor with a precision exceeding 80%. According to Fig. 8c), the RNN-based auditor and GRU-based auditor show an upward trend in recalls. Herein, both of their recalls exceed 80% when the training set size is larger than 500. The recall trends for the LSTM-based auditor and the Combined Auditor follows the opposite trend as that of GRU and RNN-based auditors. In general, the RNN-based auditor performs well across all three metrics. The LSTM-based auditor shows an excellent precision, while the GRU-based auditor obtains the highest accuracy.

> The algorithm selected for the shadow model will influence the auditor's performance. The Combined Auditor can achieve accuracy higher than the average, only if its training set is relatively small.

4.6 Effect of Noisy Queries

An evaluation of the user-level audio auditor's robustness, when provided with noisy audios, is also conducted. We also consider the effect of the noisy audios when querying different kinds of auditors trained on different shadow model architectures. We shall describe the performance of the auditor with two metrics — precision and recall; these results are illustrated in Fig. 9.
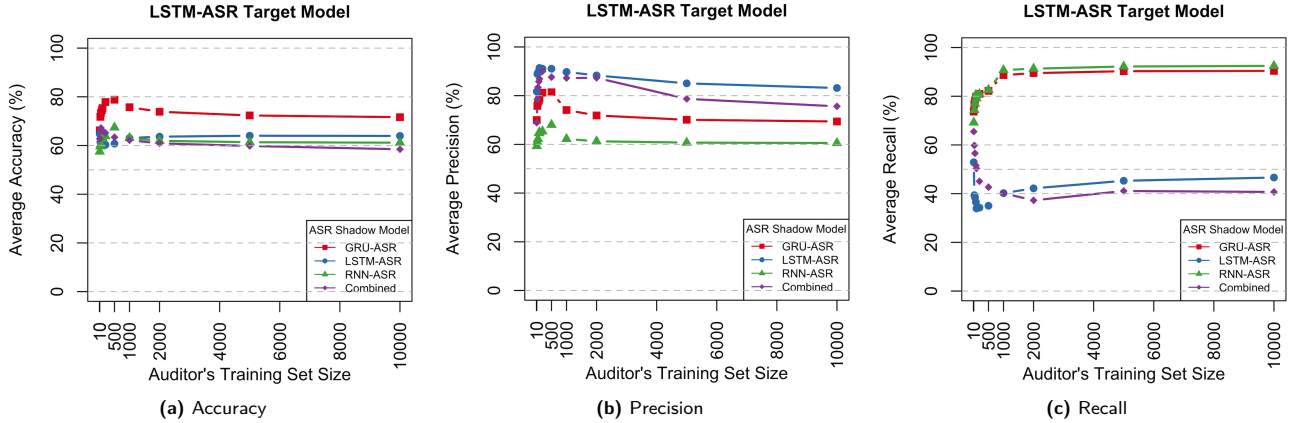
Fig. 8. Different auditor model performance when trained with different ASR shadow model architectures.

To explore the effect of noisy queries, we assume that our target model is trained with noisy audios. Under the strict black-box access to this target model, we shall use different neural network structures to build the target model(s) and the shadow model(s). That is, the target model is an LSTM-ASR target model, while the GRU-ASR shadow model is used to train the GRU-based auditor. For evaluating the effect of the noisy queries, two target models are prepared using (i) clean audios (100 hours) and (ii) noisy audios (500 hours) as training sets. In addition to the GRU-based auditor, another two auditors are constructed, an LSTM-based auditor and an RNN-based auditor. The target models audited by the latter two auditors are the same as the GRU-based auditor. Herein, the LSTM-based auditor has an LSTM-ASR shadow model whose acoustic model shares the same algorithm as its LSTM-ASR target model.

Fig. 9a and Fig. 9d compare the precision and recall of the GRU-based auditor on target models trained with clean and noisy queries, respectively. Overall, the auditor's performance drops when auditing noisy queries, but the auditor still outperforms the random guess (>50%). By varying the size of the auditor's training set, we observe the precision of the auditor querying clean and noisy audios displaying similar trends. When querying noisy audios, the largest change in precision is ≈11%, where the auditor's training set size was 500. Its precision results of querying clean and noisy audios are around 81% and 70%, respectively. However, the trends of the two recall results are fairly the opposite, and noisy queries' recalls are decreasing remarkably. The lowest decent rate of the recall is about 42%, where the auditor was trained with ten training samples. Its recall results of querying two kinds of audios are around 74%

and 32%. In conclusion, we observe the impact of noisy queries on our auditor is fairly negative.

Fig. 9b and Fig. 9e display the LSTM-based auditor's precision and recall, respectively, while Fig. 9c and Fig. 9f illustrate the RNN-based auditor's performance. Similar trends are observed from the earlier precision results in the RNN-based auditor when querying clean and noisy queries. However, curiously, the RNN-based auditor, when querying noisy audios, slightly outperforms queries on clean audios. Similar to the noisy queries effect on GRU-based auditor, the noisy queries' recall of the RNN-based auditor decreases significantly versus the results of querying the clean audios. Though noisy queries show a negative effect, all recalls performed by the RNN-based auditor exceed 50%, the random guess. As for the effect of noisy queries on the LSTM-based auditor, unlike GRU-based auditor and RNN-based auditor, the LSTM-based auditor demonstrates high robustness on noisy queries. For the most results of its precision and recall, the differences between the performance on clean and noisy queries are no more than 5%.

> In conclusion, noisy queries create a negative effect on our auditor's performance. Yet, if the shadow ASR model and the target ASR model are trained with the same algorithm, the negative effect can be largely eliminated.

4.7 Effect of Different ASR Model Pipelines on Auditor Performance

Aside from the ASR model's architecture, we examine the user-level auditor's robustness on different pipelines commonly found in ASR systems. In this section, the
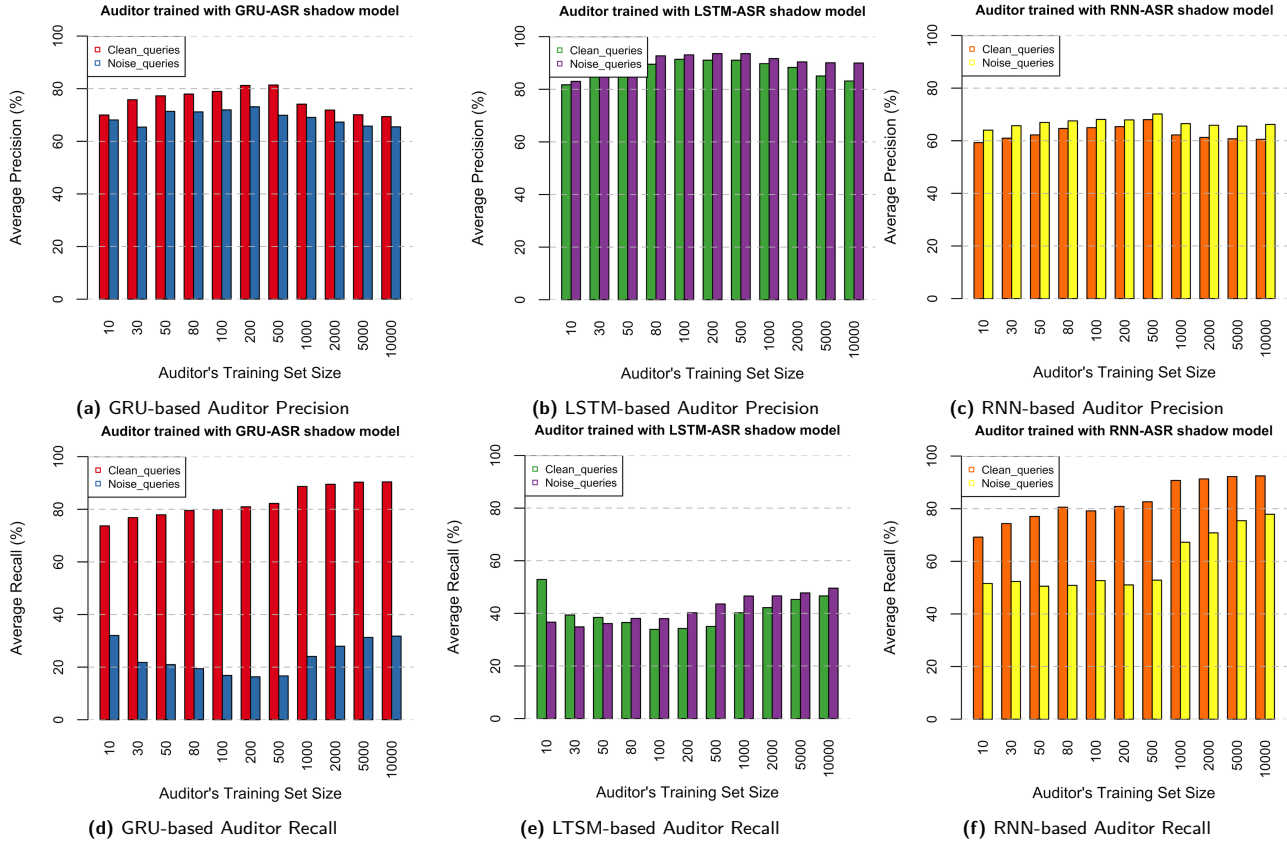
Fig. 9. Different auditor audits noisy queries with different ASR shadow model.

ASR pipeline is not just a machine learning, instead a complicated system, as shown in Fig. 1. In practice, the two most popular pipelines adopted in ASR systems are: a hybrid ASR system, or an end-to-end ASR system. We build our auditor using the GRU-ASR shadow model with two target models trained on systems built on the aforementioned ASR pipelines. Specifically, one target model utilizes the Pytorch-Kaldi toolkit to construct a hybrid DNN-HMM ASR system, while the other target model employs an end-to-end ASR system.

Fig. 10 reports the performance (accuracy, precision, and recall) of the auditor when auditing the two different target pipelines. Overall, the auditor behaves well over all metrics when auditing either target models (all above 50%). The auditor always demonstrates good performance when using a small number of training samples. The auditor targeting hybrid ASR in comparison to the end-to-end ASR target achieves a better result. A possible reason is that our auditor is constituted with a shadow model which has a hybrid ASR architecture. When focusing on the accuracy, the highest audit score of the hybrid ASR target model is 78.8%, while that of the end-to-end ASR target model is 71.92%. The

difference in the auditor's precision is not substantially, with their highest precision scores as 81.4% and 79.1%, respectively. However, in terms of recall, the auditor's ability to determine the user-level membership on the hybrid ASR target model is much higher than the end-to-end target model, with maximum recall of 90% and 72%, respectively.

When auditing the hybrid ASR target model we observed the model significantly outperforming other models. The training and testing data for both state-of-the-art ASR model architectures (i.e., hybrid and end-to-end) are the same. Thus, to confidently understand the impact of different ASR model pipelines on the auditor's performance, we shall also investigate the difference between the overfitting level of the hybrid ASR target model and that of the end-to-end ASR target model, as the overfitting of the model increases the success rate of membership inference attacks [39]. Recall that overfitting was previously defined in Section 3.3. The overfitting value of the hybrid ASR target model is measured as 0.04, while the overfitting level of the end-to-end ASR target model is 0.14. Contrary to the conclusions observed by [32], the target model that was more overfit
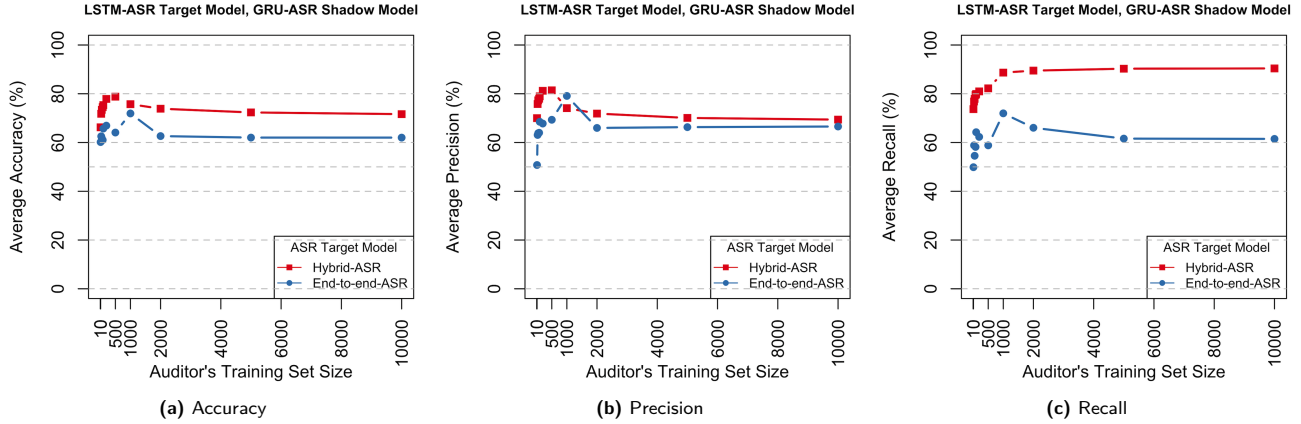
Fig. 10. The audit model audits different target ASR models trained with different pipelines.

did not increase the performance of our user-level audio auditor. One likely reason is that our auditor audits the target model by considering user-level information under strict black-box access. Compared to conventional black-box access in [32], our strict black-box access obtains its output from the transcribed text alone; consequently the influence of overfitting on specific words (WER) would be minimized. Thus, we can observe that our auditor's success is not entirely attributed to the degree of the target ASR model's overfitting alone.

> In conclusion, different ASR pipelines between the target model and the shadow model negatively impact the performance of the auditor. Nevertheless, our auditor still performs well when the target model is trained following a different pipeline (i.e., an end-to-end ASR system), significantly outperforming random guesses (50%).

4.8 Real-World Audit Test

To test the practicality of our model in the real world, we keep our auditor model locally and conduct a proof-of-concept trial to audit iPhone Siri's speech-to-text service. We select the auditor trained by the GRU shadow model with LibriSpeech 360-hour voice data as its training set. To simplify the experiments, we sample five audios per user for each user's audit. According to the results presented in Fig. 6 and Fig. 7, we select the auditor trained with five audios per user, where 1,000 users were sampled randomly as the auditor's training set. To gain the average performance of our auditor in the real world, we stored 100 auditors under the same settings

with the training set constructed 100 times. The final performance is the average of these 100 auditors' results. **Testbed and Data Preprocess.** The iPhone Siri provides strict black-box access to users, and the only dictation result is its predicted text. All dictation tasks were completed and all audios were recorded in a quiet surrounding. The clean audios were played via a Bluetooth speaker to ensure Siri can sense the audios. User-level features were extracted as per Section 3.2. The Siri is targeted on iPhone X of iOS 13.4.1.

Ground Truth. We target a particular user $\hat{u}$ iPhone Siri's speech-to-text service. From Apple's privacy policy of Siri (see Appendix D), iPhone user's Siri's recordings can be selected to improve Siri and dictation service in the long-term (for up to two years). We do note that this is an opt-in service. Simply put, this user can be labeled as "member". As for the "nonmember" user, we randomly selected 52 speakers from LibriSpeech dataset which was collected before 2014 [25]. As stated by the iPhone Siri's privacy policy, users' data "may be retained for up to two years". Thus, audios sampled from LibriSpeech can be considered out of this Siri's training set. We further regard the corresponding speakers of LibriSpeech as "nonmember" users. To avoid nonmember audios entering Siri's training data to retrain its ASR model during the testing time, each user's querying audios were completed on the same day we commenced tests for that user, with the Improve Siri & Dictation setting turned off.

As we defined above, a "member" may make the following queries to our auditor: querying the auditor (i) with audios within the target model's training set ($D_{\hat{u}} = \bigcup A^{in}_{mem}$); (ii) with audios out of the target model's training set ($D_{\hat{u}} = \bigcup A^{out}_{mem}$); (iii) with part of his or her audios within the target model's training set ($D_{\hat{u}} =$

$(\bigcup A_{mem}^{in}) \bigcup (\bigcup A_{mem}^{out}))$. Thus, we generate six "member" samples where the audios were all recorded by the target iPhone's owner, including $D_{\hat{u}} = \bigcup_{k=5} A_{mem}^{in}$, $D_{\hat{u}} = \bigcup_{m=5} A_{mem}^{out}$, and $D_{\hat{u}} = (\bigcup_{k} A_{mem}^{in}) \bigcup (\bigcup_{m} A_{mem}^{out})$, where $k = 1, 2, 3$ and $k + m = 5$. In total, we collected 58 user-level samples with 6 "member" and 52 "nonmember" samples.

Results. We load 100 auditors to test those samples and the averaged overall accuracy as 89.76%. Specifically, the average precision of predicting the "member" samples is 58.45%, while the average precision of predicting the "nonmember" samples is 92.61%. The average ROC AUC result is 72.6%, which indicates our auditor's separability in this experiment. Except for the different behaviors of Siri translating the audios from "member" and "nonmember" users, we suspect that another reason of high precision on "nonmember" is due to the LibriSpeech audios are out of Siri's dictation scope. As for the low precision rate on "member" samples, we single out the data $D_{\hat{u}} = \bigcup_{k=5} A_{mem}^{in}$ for testing. Additionally, its average accuracy result can reach 100%; thus, the auditor is much more capable in handling A_{mem}^{in} than A_{mem}^{out}, corroborating our observation in Section 4.2.

> In conclusion, our auditor shows a generally satisfying performance for users auditing a real-world ASR system, Apple's Siri on iPhone.

5 Threats to Auditors' Validity

Voiceprints Anonymization. In determining the user-level membership of audios in the ASR model, our auditor relies on the target model's different behaviors when presented with training and unseen samples. The auditor's quality depends on the diverse responses of the target model when translating audio from different users. The feature is named users' voiceprints. The voiceprint is measured in [28] based on a speaker recognition system's accuracy. Our auditor represents the user's voiceprint according to two accumulated features, including missing characters and extra characters. However, if an ASR system is built using voice anonymization, our user-level auditor's performance would degrade significantly. The speaker's voice is disguised in [28] by using robust voice conversation while ensuring the correctness of speech content recognition. Herein, the most popular technique of voice conversation is frequency

warping [43]. In addition, abundant information about speakers' identities is removed in [42] by using adversarial training for the audio content feature. Fig. 1 shows that the average accuracy of the auditor dropped by approximately 20% without using the two essential features. Hence, auditing user-level membership in a speech recognition model trained with anonymized voiceprints remains as a future avenue of research.

Differentially Private Recognition Systems. Differential privacy (DP) is one of the most popular methods to prevent ML models from leaking any training data information. The work [40] protects the text generative model by applying user-level DP to its language model. This method contains a language model during the hybrid ASR system's training, on which the user-level DP can be applied to obscure the identity, at the sacrifice of transcription performance. The speaker and speech characterization process is protected in [23] by inserting noise during the learning process. However, due to strict black-box access and the lack of output probability information, our auditor's performance remains unknown on auditing the ASR model with DP. The investigation of our auditor's performance to this user protection mechanism is open for future research.

Workarounds and Countermeasures. Although Salem et al. [32] has shown neither the shadow model nor the attack model are required to perform membership inference, due to the constraints of strict black-box access, the shadow model and auditor model approach provide a promising means to perform a more difficult task of user-level membership inference. Instead of the output probabilities, we mainly leverage the ASR model's translation errors in the character level to represent the model's behaviors. Alternative countermeasures against the membership inference, such as dropout, generally change the target model's output probability. However, the changes to probabilities of the ASR model's output are not as sensitive as changes to its translated text [32]. Studying the extent of this sensitivity of ASR models remains as our future work.

Synthetic ASR Models. Another limitation of our work is that we evaluate our auditor on synthetic ASR systems trained on real-world datasets, and we have not applied the auditor to extensive set of real-world models aside from Siri. However, we believe that our reconstruction of the ASR models closely mirrors ASR models in the wild.

6 Related Work

Membership Inference Attacks. The attack distinguishes whether a particular data sample is a member of the target model's training set or not. Traditional membership inference attacks against ML models under black-box access leverage numerous shadow models to mimic the target model's behavior [10, 17, 39]. Salem et al. [32] revealed that membership inference attacks could be launched by directly utilizing the prediction probabilities and thresholds of the target model. Both works [17] and [49] prove that overfitting of a model is sufficient but not a necessity to the success of a membership inference attack. Yeom et al. [49] as well as Farokhi and Kaafar [8] formalize the membership inference attack with black-box and white-box access. All previously mentioned works consider record-level inference; however, Song and Shmatikov [40] deploy a user-level membership inference attack in text generative models, with only the top-n predictions known.

Trustworthiness of ASR Systems. The ASR systems are often deployed on voice-controlled devices [5], voice personal assistants [37], and machine translation services [6]. Tung and Shin [45] propose *SafeChat* to utilize a masking sound to distinguish authorized audios from unauthorized recording to protect any information leakage. Recent works [18] and [44] propose an audio cloning attack and audio reply attack against the speech recognition system to impersonate a legitimate user or inject unintended voices. Voice masquerading to impersonate users on the voice personal assistants has been studied [50]. Whereas Zhang et al. [50] propose another attack, namely *voice squatting*, to hijack the user's voice command, producing a sentence similar to the legal command. Du et al. [6] generate the adversarial audio samples to deceive the end-to-end ASR systems.

Auditing ML Models. Many of the current proposed auditing services also seek to audit the bias and fairness of a given model [31]. Works have also been presented to audit the ML model to learn and check the model's prediction reliability [1, 12, 36]. Moreover, the auditor is utilized to evaluate the ML model's privacy risk when protecting an individual's digital rights [21, 40].

Our Work. Our user-level audio auditor audits the ASR model under the strict black-box access. As shown in Section 3, we utilize the ASR model's translation errors in the character level to represent the model's behavior. Compared to related works under black-box access, our auditor does not rely on the target model's output probability [32, 39]. In addition, we sidestep the

feature pattern of several top-ranked outputs of the target model adopted by Song and Shmatikov [40], instead we use one text output, the user's speed, and the input audio's true transcription, as we do not have access to the output probability (usually unattainable in ASR systems). Hence, our constraints of strict black-box access only allow accessing one top-ranked output. In this case, our user-level auditor (78.81%) outperforms Song's and Shmatikov's user-level auditor (72.3%) in terms of accuracy. Moreover, Hayes et al. [10] use adversarial generative networks (GANs) to approximate the target model's output probabilities while suffering big performance penalties with only 20% accuracy, our auditor's accuracy is far higher. Furthermore, our auditor is much easier to be trained than the solution of finding outlier records with a unique influence on the target model [17], because we only need to train one shadow model instead of many shadow (or reference) models.

7 Conclusion

This work highlights and exposes the potential of carrying out user-level membership inference audit in IoT voice services. The auditor developed in this paper has demonstrated promising data transferability, while allowing a user to audit his or her membership with a query of only nine audios. Even with audios are not within the target model's training set, the user's membership can still be faithfully determined. While our work has yet to overhaul the audit accuracy on various IoT applications across multiple learning models in the wild, we do narrow the gap towards defining clear membership privacy in the user level, rather than the record level [39]. However, questions remain about whether the privacy leakage hails from the data distribution or its intrinsic uniqueness of the record. More detail about limitations are stated in Appendices. Nevertheless, as we have shown, both a small training set size and the Combined Auditor, which combines results from various ASR shadow models to train the auditor, have a positive effect on the IoT audit model; on the contrary, audios recorded in a noisy environment and different ASR pipelines impose a negative effect on the given auditor; fortunately, the auditor still outperforms random guesses (50%). Examining other performance factors on more real-world ASR systems in addition to our iPhone Siri trial and extending possible countermeasures against auditing are all worth further exploration.

Acknowledgments

References

[1] P. Adler, C. Falk, S. A. Friedler, T. Nix, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018.

[2] BBC. Hmrc forced to delete five million voice files, 2019. URL https://www.bbc.com/news/business-48150575.

[3] D. U. CCPA. California consumer privacy act (ccpa) website policy, 2020.

[4] CCTV. Beware of WeChat voice scams: "cloning" users after WeChat voice, 2018. URL https://translate.google.com/translate?hl=en&sl=zh-CN&u=https://finance.sina.com.cn/money/bank/bank_hydt/2018-11-26/doc-ihmutuec3748767.shtml&prev=search.

[5] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[6] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah. Sirenattack: Generating adversarial audio for end-to-end acoustic systems. *arXiv preprint arXiv:1901.07846*, 2019.

[7] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data and repealing Directive 95/46/EC (general data protection regulation). *Official Journal of the European Union*, 119:1–88, 2016.

[8] F. Farokhi and M. A. Kaafar. Modelling and quantifying membership information leakage in machine learning. *arXiv preprint arXiv:2001.10648*, 2020.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon Technical Report*, 93, 1993.

[10] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019.

[11] A. Hern. Apple contractors 'regularly hear confidential details' on siri recordings, 2019. URL https://www.theguardian.com/technology/2019/jul/26/apple-contractors-regularly-hear-confidential-details-on-siri-recordings.

[12] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[13] W. Kyle. How Amazon, Apple, Google, Microsoft, and Samsung treat your voice data, 2019. URL https://venturebeat.com/2019/04/15/how-amazon-apple-google-microsoft-and-samsung-treat-your-voice-data/.

[14] A. Liu, H.-y. Lee, and L.-s. Lee. Adversarial training of end-to-end speech recognition using a criticizing language model. In *Proceeding of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[15] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng. Socinf: Membership inference attacks on social media health data with machine learning. *IEEE Transactions on Computational Social Systems*, 6(5):907–921, 2019.

[16] S. Lokesh, P. K. Malarvizhi, M. D. Ramya, P. Parthasarathy, and C. Gokulnath. An automatic tamil speech recognition system by using bidirectional recurrent neural network with self-organizing map. *Neural Computing and Applications*, pages 1–11, 2018.

[17] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[18] H. Malik. Securing voice-driven interfaces against fake (cloned) audio attacks. In *Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 512–517. IEEE, 2019.

[19] E. McReynolds, S. Hubbard, T. Lau, A. Saraf, M. Cakmak, and F. Roesner. Toys that listen: A study of parents, children, and Internet-connected toys. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5197–5207. ACM, 2017.

[20] M. Mehrabani, S. Bangalore, and B. Stern. Personalized speech recognition for Internet of Things. In *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 369–374. IEEE, 2015.

[21] Y. Miao, B. Z. H. Zhao, M. Xue, C. Chen, L. Pan, J. Zhang, D. Kaafar, and Y. Xiang. The audio auditor: Participant-level membership inference in voice-based IoT. *CCS Workshop of Privacy Preserving Machine Learning*, 2019.

[22] M. Nasr, R. Shokri, and A. Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 634–646, 2018.

[23] A. Nautsch, A. Jiménez, A. Treiber, J. Kolberg, C. Jasserand, E. Kindt, H. Delgado, M. Todisco, M. A. Hmani, A. Mtibaa, et al. Preserving privacy in speaker and speech characterisation. *Computer Speech & Language*, 58:441–480, 2019.

[24] S. Nick. Amazon may give app developers access to Alexa audio recordings, 2017. URL https://www.theverge.com/2017/7/12/15960596/amazon-alexa-echo-speaker-audio-recordings-developers-data.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio

books. In *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[26] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[27] J. M. Perero-Codosero, J. Antón-Martín, D. T. Merino, E. L. Gonzalo, and L. A. Hernández-Gómez. Exploring open-source deep learning ASR for speech-to-text TV program transcription. In *Proceedings of the IberSPEECH*, pages 262–266, 2018.

[28] J. Qian, H. Du, J. Hou, L. Chen, T. Jung, and X. Li. Speech sanitizer: Speech content desensitization and voice anonymization. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[29] M. Ravanelli, T. Parcollet, and Y. Bengio. The Pytorch-Kaldi speech recognition toolkit. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6465–6469. IEEE, 2019.

[30] A. Rousseau, P. Deléglise, and Y. Esteve. Ted-lium: An automatic speech recognition dedicated corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 125–129, 2012.

[31] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018.

[32] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[33] M. Sapna. Hey, Alexa, what can you hear? and what will you do with it?, 2018. URL https://www.nytimes.com/2018/03/31/business/media/amazon-google-privacy-digital-assistants.html.

[34] P. Sarah. 41% of voice assistant users have concerns about trust and privacy, report finds, 2019. URL https://techcrunch.com/2019/04/24/41-of-voice-assistant-users-have-concerns-about-trust-and-privacy-report-finds/.

[35] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *arXiv preprint arXiv:1808.05665*, 0(0):1–18, 2018.

[36] P. Schulam and S. Saria. Can you trust this prediction? Auditing pointwise reliability after learning. *arXiv preprint arXiv:1901.00403*, 2019.

[37] F. H. Shezan, H. Hu, J. Wang, G. Wang, and Y. Tian. Read between the lines: An empirical measurement of sensitive applications of voice personal assistant systems. In *Proceedings of the Web Conference*, WWW '20. ACM, 2020.

[38] M. Shokoohi-Yekta, Y. Chen, B. Campana, B. Hu, J. Zakaria, and E. Keogh. Discovery of meaningful rules in time series. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1085–1094. ACM, 2015.

[39] R. Shokri, M. S. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017.

[40] C. Song and V. Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 196–206, 2019.

[41] L. Song, R. Shokri, and P. Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 241–257, 2019.

[42] B. M. L. Srivastava, A. Bellet, M. Tommasi, and E. Vincent. Privacy-preserving adversarial representation learning in ASR: Reality or illusion? *arXiv preprint arXiv:1911.04913*, 2019.

[43] D. Sundermann and H. Ney. Vtln-based voice conversion. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No. 03EX795)*, pages 556–559. IEEE, 2003.

[44] F. Tom, M. Jain, and P. Dey. End-to-end audio replay attack detection using deep convolutional networks with attention. In *Proceedings of the Interspeech Conference*, pages 681–685, 2018.

[45] Y.-C. Tung and K. G. Shin. Exploiting sound masking for audio privacy in smartphones. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 257–268, 2019.

[46] F. Weninger, H. Erdogan, W. S, V. E, J. Le Roux, J. R. Hershey, and B. Schuller. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation*, pages 91–99. Springer, 2015.

[47] S. Wildstrom. Nuance exec on iphone 4s, siri, and the future of speech, 2011. URL https://techpinions.com/nuance-exec-on-iphone-4s-siri-and-the-future-of-speech/3307.

[48] M. Xue, G. Magno, E. Cunha, V. Almeida, and K. W. Ross. The right to be forgotten in the media: A data-driven study. *Proceedings on Privacy Enhancing Technologies*, 2016(4): 389–402, 2016.

[49] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[50] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian. Dangerous skills: Understanding and mitigating security risks of voice-controlled third-party functions on virtual personal assistant systems. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (S&P)*, pages 1381–1396. IEEE, 2019.

Appendices

A. Datasets

The **LibriSpeech** speech corpus (LibriSpeech) contains 1,000 hours of speech audios from audiobooks which are part of the LibriVox project [25]. This corpus is famous in training and evaluating speech recognition systems. At least 1,500 speakers have contributed their voices to this corpus. We use 100 hours of clean speech data with 29,877 recordings to train and test our target model. 360 hours of clean speech data, including 105,293 recordings, are used for training and testing the shadow models. Additionally, there are 500 hours of noisy data used to train the ASR model and to test our auditor's performance in a noisy environment.

The **TIMIT** speech corpus (TIMIT) is another famous speech corpus used to build ASR systems. This corpus recorded audios from 630 speakers across the United States, totaling 6,300 sentences [9]. In this work, we use all this data to train and test a target ASR model, and then audit this model with our auditor.

The **TED-LIUM** speech corpus (TED) collected audios based on TED Talks for ASR development [30]. This corpus was built from the TED talks of the IWSLT 2011 Evaluation Campaign. There are 118 hours of speeches with corresponding transcripts.

B. Evaluation Metrics

The user-level audio auditor is evaluated with four metrics calculated from the confusion matrix, which reports the number of true positives, true negatives, false positives and false negatives: **True Positive (TP)**, the number of records we predicted as "member" are correctly labeled; **True Negative (TN)**, the number of records we predicted as "nonmember" are correctly labeled; **False Positive (FP)**, the number of records we predicted as "member" are incorrectly labeled; **False Negative (FN)**, the number of records we predicted as "nonmember" are incorrectly labeled. Our evaluation metrics are derived from the above-mentioned numbers.

- Accuracy: the percentage of records correctly classified by the auditor model.
- Precision: the percentage of records correctly determined as "member" by the auditor model among all records determined as "member".
- Recall: the percentage of all true "member" records correctly determined as "member".

- F1-score: the harmonic mean of precision and recall.

C. ASR Models' Architectures

On the LibriSpeech 360-hour voice dataset, we build one GRU-ASR model with the Pytorch-Kaldi toolkit. That is, we train a five-layer GRU network with each hidden layer of size 550 and one Softmax layer. We use `tanh` as the activation function. The optimization function is Root Mean Square Propagation (RMSProp). We set the learning rate as 0.0004, the dropout rate for each GRU layer 0.2, and the number of epochs of training 24.

On the LibriSpeech 360-hour voice dataset, we train another ASR model using the Pytorch-Kaldi toolkit. Specifically, it is a four-layer RNN network with each hidden layer of size 550 using ReLU as the activation function and a Softmax layer. The optimization function is RMSProp. We set the learning rate as 0.00032, the dropout rate for each RNN layer 0.2, and the number epochs of training 24.

On the LibriSpeech 360-hour voice dataset, we train one hybrid LSTM-ASR model. The acoustic model is constructed with a four-layer LSTM and one Softmax layer. The size of each hidden LSTM layer is 550 along with 0.2 dropout rate. The activation function is `tanh`, while the optimization function is RMSProp. The learning rate is 0.0014, and the maximum number of training epochs is 24.

On the LibriSpeecch 100-hour voice dataset, we train a hybrid ASR model. The acoustic model is constructed with a four-layer LSTM and one Softmax layer. Each hidden LSTM layer has 550 neurons along with 0.2 dropout rate. The activation function is `tanh`, while the optimization function is RMSProp. The learning rate is 0.0016, and the maximum number of training epochs is 24.

On the LibriSpeecch 100-hour voice dataset, we train an end-to-end ASR model. The encoder is constructed with a five-layer LSTM with each layer of size 320 and with 0.1 dropout rate. We use one layer location-based attention with 300 cells. The decoder is constructed with a one-layer LSTM with 320 neurons along with 0.5 dropout rate. The CTC decoding is enabled with a weight of 0.5. The optimization function is Adam. The learning rate is 1.0, and the total number of training epochs is 24.

On the TEDLium dataset, we train a hybrid ASR model. The acoustic model is constructed with a four-layer LSTM and one Softmax layer. Each hidden LSTM layer has 550 neurons along with 0.2 dropout rate. The

activation function is tanh, while the optimization function is RMSProp. The learning rate is 0.0016, and the number of maximum training epochs is 24.

On the TIMIT dataset, we train a hybrid ASR model. The acoustic model is constructed with a four-layer LSTM and one Softmax layer. Each hidden LSTM layer has 550 neurons along with 0.2 dropout rate. The activation function is tanh, while the optimization function is RMSProp. The learning rate is 0.0016, and the number of maximum training epochs is 24.

D. Real-World Audit Test

Siri is a virtual assistant provided by Apple in their iOS, iPadOs, and macOS. Siri's natural language interface is used to answer users' voice queries and make recommendations [47]. The privacy policy of Apple's Siri is shown in Fig. 11. The user, who is considered as a member user of Siri's ASR model in our setting, has used the targeted iPhone for more than two years, frequently interacting with Siri, often with the Improve Siri & Dictation service opted in. As for the member user's member audio, we carefully chose five phrases that the user had certainly used when engaging with Siri. Starting with "Hey Siri" and phrases from common interactions including "Hey Siri", "What's the weather today", "What date is it today", "Set alarm at 10 o'clock", and "Hey Siri, what's your name". As for the member user's non-member audio, we chose five short phrases in LibriSpeech that the user had never used to interact with Siri (e.g. "we ate at many men's tables uninvited"). These phrases were recorded using the member user's voice along with our member user's non-member audios. The use of either set of phrases should produce an ability to audit as recall that our method imitates the user as a whole when auditing the model, irrespective of whether a specific audio phrase was used to train/update this model. Lastly, for the nonmember users' nonmember audio, the target Siri's language is English (Australia). Since the LibriSpeech dataset was collected before 2014 and as stated by the iPhone Siri's privacy policy, users' data "may be retained for up to two years". We consider these recordings to not be part of the Siri training dataset, and thus we have nonmember users' nonmember audio. We further assume that our selected user phrases about book reading are nonmembers.
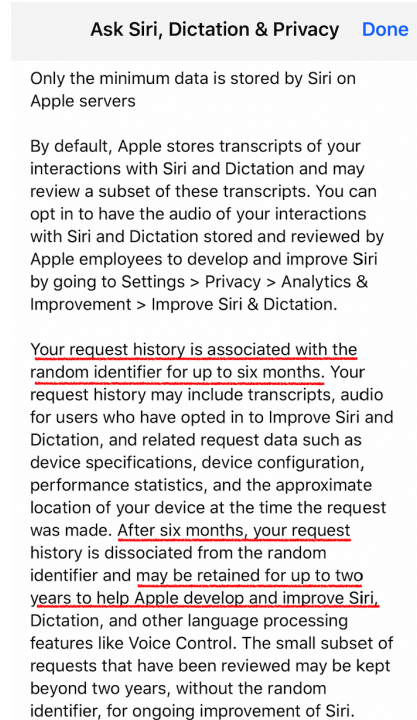

Ask Siri, Dictation & Privacy Done

Only the minimum data is stored by Siri on Apple servers

By default, Apple stores transcripts of your interactions with Siri and Dictation and may review a subset of these transcripts. You can opt in to have the audio of your interactions with Siri and Dictation stored and reviewed by Apple employees to develop and improve Siri by going to Settings > Privacy > Analytics & Improvement > Improve Siri & Dictation.

Your request history is associated with the random identifier for up to six months. Your request history may include transcripts, audio for users who have opted in to Improve Siri and Dictation, and related request data such as device specifications, device configuration, performance statistics, and the approximate location of your device at the time the request was made. After six months, your request history is dissociated from the random identifier and may be retained for up to two years to help Apple develop and improve Siri, Dictation, and other language processing features like Voice Control. The small subset of requests that have been reviewed may be kept beyond two years, without the random identifier, for ongoing improvement of Siri.

Fig. 11. The privacy policy of Apple's Siri

E. Limitations and Future Work

Further Investigation on Features. From our set of selected features both audio-specific features and features capturing model behaviors performs well, as observed in our results. It remains to be seen if additional audio-specific features would specifically aid the task of user-level auditing. As there are many potential feature candidates, we consider this as part of future work.

Auditing Performance with Varied Numbers of Queries. In our auditor, we observe that only a limited number of queries per user is necessary to audit the target ASR model, especially when the auditor is trained with a limited audios per user. An interesting observation was that our user-level auditor's recall performance on more queries per user declines under the strict black-box access. We are continuing our investigation into why our auditor's ability to find unauthorized use of user data varies in this manner when being queried with different numbers of audios.

Member Audio in Siri Auditing. In our setting, we make our best effort to ensure member audios are used for training. However, in our real-world evaluation, even with the "Improve Siri & Dictation" setting turned on, with an extended period of continual use by our user, we cannot guarantee that member audios of our member user where actually used for training although we are confident they have been included.