# Evaluating the Vulnerability of End-to-End Automatic Speech Recognition Models To Membership Inference Attacks

*Muhammad A. Shah, Joseph Szurley, Markus Mueller, Athanasios Mouchtaris, Jasha Droppo*

Alexa Machine Learning, Amazon, USA

{msshahmz,jszurley,mumarkus,mouchta,drojasha}@amazon.com

## Abstract

Recent studies have shown that it may be possible to determine if a machine learning model was trained on a given data sample, using Membership Inference Attacks (MIA). In this paper we evaluate the vulnerability of state-of-the-art speech recognition models to MIA under black-box access. Using models trained with standard methods and public datasets, we demonstrate that without any knowledge of the target model's parameters or training data a MIA can successfully infer membership with precision and recall more than 60%. Furthermore, for utterances from about 39% of the speakers the precision is more than 75%, indicating that training data membership can be inferred more precisely for some speakers than others. While strong regularization reduces the overall accuracy of MIA to almost 50%, the attacker can still infer membership for utterances from 25% of the speakers with high precision. These results indicate that (1) speaker-level MIA success should be reported, along with overall accuracy, to provide a holistic view of the model's vulnerability and (2) conventional regularization is an inadequate defense against MIA. We believe that the insights gleaned from this study can direct future work towards more effective defenses.

**Index Terms**: speech recognition, privacy, security, membership inference

## 1. Introduction

Modern machine learned models rely on massive amounts of training data that, in commercial settings, is often crowdsourced from customers. In order to secure customers' data and maintain its privacy, entities that build these models rely on strict data handling policies. While such policies are effective during model building, they alone, however, may not provide sufficient guarantees *after* the model has been deployed [1].

Recent studies have shown that a machine learned model may reveal information about its training data when subjected to a Membership Inference Attack (MIA) [1, 2, 3, 4]. A MIA attempts to determine if a given data point was used to train a given model [1]. Successful MIAs are possible because the model may fit its training data too closely causing its parameters to encode peculiarities of the training data. Consequently, the model's outputs in response to training and non-training data differ. Therefore, by analysing the model's parameters [5] and/or outputs [1] the membership of a data point can be inferred.

Despite their ubiquitous deployment [6, 7], the vulnerability of ASR models to MIA has not been thoroughly evaluated. To the best of our knowledge, only Miao et al [4] have studied the vulnerability of ASR models but their analysis is limited to *speaker-level* MIA, which reveals if a user's data was used for training, but does not indicate if the model leaks information about the *content* of the user's speech, which may be sensitive.

We thank Abhinav Aggarwal and Grant Strimel for their valuable feedback.

To fill the gap in the current literature, in this paper we present an empirical study that evaluates the vulnerability of state-of-the-art ASR models to MIA. We assume a black-box threat model i.e. the attacker sees only transcripts and the likelihoods of the $k$-best hypotheses, and has no knowledge of the model parameters. Similar to [8, 3], we use the MIA accuracy, i.e. how accurately can the MIA determine if a point was in the training set, as a measure of the vulnerability.

Our results show that even with black box access to the model and no knowledge of the training data distribution, the overall MIA accuracy is more than 60%, which is better than chance. We observe that MIA primarily uses Word Error Rate (WER) to infer membership – utterances that yield lower WER tend to be in the training set. Furthermore, we make a novel discovery that *training data membership can be inferred more precicely for some speakers than others*. Specifically, we find that for about 39% of the speakers in the test set the MIA can infer membership with greater than 75% precision. To the best of our knowledge, past literature [8, 9] has only shown that it may be easier to infer membership of some outlying data points, but, this analysis has not been conducted at the user level.

Since the MIA exploits the difference in WER on training and non-training data, we consider regularization as a defense. We find that the overall MIA accuracy drops to 52% against a strongly regularized model, but the MIA is able to predict membership for 25.3% of the speakers with precision greater than 75%. This implies that (1) regularization is not sufficient to mitigate the risk of MIA for all users and (2) speaker-level MIA success rate is a more accurate measure of an ASR model's vulnerability to MIA than overall accuracy.

## 2. Background

### 2.1. Membership Inference Attacks on ML Models

The MIA task [1] is defined as follows. Consider a universe, $\mathcal{U} : \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the set of data samples (human speech in our case) and $\mathcal{Y}$ is the set the corresponding labels (transcripts in our case). There is a service provided called Alice who has access to a dataset $\mathcal{D}_A : \mathcal{X}_A \times \mathcal{Y}_A \subset \mathcal{U}$. It uses $\mathcal{D}_A^{train} \subset \mathcal{D}_A$ to train a ML model, $g_{target} : \mathcal{X} \rightarrow \mathcal{Y}$, which we call the *target model*. Alice has an adversary called, Bob, who has two datasets $\mathcal{D}_B, \mathcal{D}_{test} \subset \mathcal{U}$ and wants to determine $\mathcal{D}_{test} \cap \mathcal{D}_A^{train}$. To this end, Bob trains an *attack model*, $m$, perhaps using side-information from $\mathcal{D}_B$ (see 3.3 for details), such that for $x, y \in \mathcal{U}$, $m(x, y, g_{target}) = \mathbb{1}\left[x, y \in \mathcal{D}_A^{train}\right]$ and applies it to $\mathcal{D}_{test}$.

Shokri et al [1] introduced MIA for deep neural networks and proposed an attack that exploits the difference in the class probabilities returned by $g_{target}$ in response to data from inside and outside the training set. Subsequently, Salem et al [10] showed that MIA can succeed even if the attacker does not have access to the distribution of $\mathcal{D}_A$, while Yeom et al [11] showed that a naive attack model, $m(x, y, g_{target}) = \mathbb{1}[g_{target}(x) =$

$y]$ can yield non-trivial accuracy *if the generalization gap is large enough*. Song et al [12] and Choquette-Choo et al [8] propose attacks that use adversarial perturbations, however [12] requires the full confidence vector, while [8] does not. While these studies were conducted on classification models, others have been conducted on sequence generation tasks like machine translation [3], text generation [13] and ASR [4]. It is worth noting that [4] considers only speaker level attacks, which are different from the utterance level attacks considered in our study.

Defenses against MIA range from standard regularization techniques like Dropout or L2 regularization [1, 10] to perturbing the output probabilities [1, 14, 15]. Unfortunately the latter fails entirely in the face of label only attacks such as [8, 11]. Heavy regularization reduces the generalization error and the success of MIA *in the average case*, however, it has been shown in [9] and 4.4.3 that the membership of certain data points can still be predicted with high precision. We show that this is because the model overfits to some training samples more than others.

### 2.2. RNN-Transducer

In our experiments we use the RNN-Transducer (RNN-T) ASR model, which is an end-to-end neural sequence-to-sequence architecture that has been widely used for ASR [16, 17, 18]. The RNN-T consists of an encoder, a decoder and a joint network, which are trained as follows. The encoder is a RNN that takes as input an audio signal, $\mathbf{x} = x_1, x_2, ..., x_T$ that has been preprocessed into $T$ frames, and produces a feature representation, $h^{enc} \in \mathbb{R}^{T\times D}$. The decoder is also a RNN that receives the ground truth symbol sequence, $\mathbf{y}^* = y_1^*, y_2^*, ..., y_U^*$, where each $y_u^*$ is part of a vocabulary $\mathcal{V}$ of $K$ symbols, and produces a feature representation, $h^{dec} \in \mathbb{R}^{U\times D}$. Finally the joint network takes as input $h^{enc}$ and $h^{dec}$ and returns a probability lattice $\mathbf{P} \in \mathbb{R}^{T\times U\times K+1}$ over the vocabulary plus the blank symbol, such that $\mathbf{P}_{tuk} = P(y_u = k|x_1, ..., x_t, y_0^*, ..., y_{u-1}^*)$. Each path through $\mathbf{P}$ corresponds to an *alignment* between the audio frames and the output sequence. The probability of an alignment can be computed as the product of the probabilities of the transitions that comprise the alignment. Summing the probabilities of all the alignments corresponding to $\mathbf{y}^*$ gives the posterior probability $P(\mathbf{y}^*|\mathbf{x})$. During training the model parameters are optimized to minimize the negative log of the posterior. The inference procedure is similar except that the decoder computes $h_u^{dec}$ for $y_u$, based on the previously emitted symbol, $y_{u-1}$, instead of the ground truth. The joint network receives $h_t^{enc}$ and $h_u^{dev}$ as input, where $t$ is the number of blanks the model has emitted until now, and computes $P(y_u|x_t, y_{u-1})$ from which the next output symbol is sampled.

## 3. Methodology

### 3.1. Threat Models

A threat model is defined along two criteria – the level of access Bob has to $g_{target}$ and the information Bob has about its training data, $\mathcal{D}_A^{train}$. We assume Bob has black box access to $g_{target}$, i.e. he has no knowledge about $g_{target}$'s architecture or parameters. However, this assumption does not preclude Bob trying to guess $g_{target}$'s architecture based on published literature. Furthermore, Bob can only interact with $g_{target}$ by querying it with utterances in response to which $g_{target}$ returns only the transcripts, $\tilde{Y} \in \mathcal{Y}^k$ and the Negative Log-Likelihoods (NLL), $-\ln P(\mathbf{y} \in \tilde{Y}|\mathbf{x})$ (see 2.2), for the $k$-best hypotheses.

With regards to access to $\mathcal{D}_A^{train}$, we consider three scenarios which we refer to as no, partial and full knowledge. In the *no knowledge (NK)* scenario Bob has no information about $\mathcal{D}_A$, i.e. $\mathcal{D}_B \cap \mathcal{D}_A = \emptyset$. Whereas in the *partial knowledge (PK)* scenario Bob knows the distribution of the data in $\mathcal{D}_A^{train}$ but he does not have access to the exact set of utterances, i.e. $\mathcal{D}_B \subset \mathcal{D}_A$ and $\mathcal{D}_B \cap \mathcal{D}_A^{train} = \emptyset$. Finally, in the *full knowledge (FK)* scenario, Bob knows exactly the training data used by Alice for training $g_{target}$ i.e. $\mathcal{D}_B = \mathcal{D}_A^{train}$. This may be contrived scenario, but we include it in our analysis to determine an upper-bound on the Bob's success rate under black box model access. As mentioned in 2.1, after training his attack model, $m$, on $\mathcal{D}_B$, Bob will attempt to infer the membership of utterances in $\mathcal{D}_{test}$. The set $\mathcal{D}_{test}$ is constructed such that $\mathcal{D}_{test} = \mathcal{D}_A^{test} \cup \tilde{\mathcal{D}}$, where $\mathcal{D}_A^{test} \subset \mathcal{D}_A^{train}$, $\tilde{\mathcal{D}} \subset \mathcal{U}$, and $\tilde{\mathcal{D}} \cap (\mathcal{D}_A^{train} \cup \mathcal{D}_B) = \emptyset$. Note that Bob does not have any information about $\mathcal{D}_{test} \cap \mathcal{D}_A^{test}$ so he can not use $\mathcal{D}_{test}$ to train $m$, however, we (the authors) know $\mathcal{D}_{test} \cap \mathcal{D}_A^{test}$ so we can determine how accurate $m$ is.

### 3.2. Feature Selection

Since ML models tend to overfit to their training data, Bob expects $g_{target}$ to have a *generalization gap*, i.e. $g_{target}$ transcribes an *inset* utterance ($x \in \mathcal{D}_A^{train}$) more accurately than an *outset* utterance ($x \notin \mathcal{D}_A^{train}$). To exploit this gap, Bob uses the Word Error Rate (WER) [19] between each hypothesis, $y \in \tilde{Y}$, and the ground truth, $y^*$, as his primary feature.

However, WER is a very coarse feature which obfuscates the influence of several other factors that may impact the transcription accuracy. For instance, longer utterances can be challenging for ASR models because the number of possible alignments in $\mathbf{P}$ (see 2.1), increases exponentially with the length of the utterance. Therefore, the WER on longer utterances may be higher than the WER on shorter utterances regardless of their training data membership. To account for the effect of utterance length Bob includes the lengths of the reference and hypotheses transcripts, and their ratio as features. It is also possible that the model would to make different types of errors on inset and outset utterances, so Bob includes the number of insertions, deletions and substitutions required to transform $y \in \tilde{Y}$ to $y^*$. These features along with the WERs for the $k$-best hypotheses, comprise a feature set of size $6k + 1$ that we refer to as *wer+lens*.

Taking *wer+lens* to be his canonical feature set Bob augments it with several other features. Expecting the model to make more confident prediction on inset data, he adds the NLL of each $y \in \tilde{Y}$ to obtain *wer+lens+nll*. To test the hypothesis that the model may be more accurate at predicting certain words than others he adds binary features for each inserted, deleted and substituted word and obtain *wer+lens+errors*. To test the hypothesis that $g_{target}$ is more adept at transcribing audio signal with particular characteristics, Bob adds the mean, variance and kurtosis of the frame-wise feature vectors of the audio signal to the feature set to obtain *wer+lens+audioStats*.

### 3.3. Attack Model Training Protocol

To train the attack model, $m$, Bob follows the shadow model protocol from [1], which is described in Algorithm 1. Bob splits $\mathcal{D}_B$ into $\mathcal{D}_B^1$ and $\mathcal{D}_B^0$, and trains an ASR model, $g_{proxy}$ on $\mathcal{D}_B^1$ (see 4.1.2 for details). He then queries $g_{proxy}$ with $\mathcal{D}_B$ and computes features $\mathcal{F}_B$ from the model's outputs. He normalizes the features to zero mean and one standard deviation and splits them into $\mathcal{F}_B^1$ and $\mathcal{F}_B^0$ such that $\mathcal{F}_B^i$ contains features extracted from $\mathcal{D}_B^i$. After assigning label $i$ to $f \in \mathcal{F}_B^i$, Bob combines $\mathcal{F}_B^1$ and $\mathcal{F}_B^0$ into $\mathcal{D}_{attack}$, which he uses to train a binary classifier $C$. The attack model is obtained by pipelining querying, feature extraction and classification into a single function, `isMember`.

**Algorithm 1:** Attack model training protocol

**Function** $\texttt{trainAttackClassifier}(\mathcal{D}_B)$:
 $\mathcal{D}_B^1, \mathcal{D}_B^0 \leftarrow \texttt{split}(\mathcal{D}_B)$
 $g_{proxy} \leftarrow \texttt{trainASRModel}(\mathcal{D}_B^1)$
 $\mathcal{F}_B \leftarrow \texttt{extractFeatures}(g_{proxy}, \mathcal{D}_B)$
 $\mathcal{F}_B^1, \mathcal{F}_B^0 \leftarrow \texttt{slpit}(\texttt{normalize}(\mathcal{F}_B))$
 $\mathcal{D}_{attack} \coloneqq [(f, \mathbb{1}(f \in \mathcal{F}_B^1) | \forall f \in \mathcal{F}_B^1 + \mathcal{F}_B^0]$
 $C \leftarrow \texttt{trainBinaryClassifier}(\mathcal{D}_{attack})$
 **return** $C$
**Function** $\texttt{isMember}(\mathcal{D}_{eval}, g_t, C)$:
 $\mathcal{F}_{eval} \leftarrow \texttt{extractFeatures}(g_t, \mathcal{D}_{eval})$
 $\tilde{\mathcal{F}}_{eval} \leftarrow \texttt{normalize}(\mathcal{F}_{eval})$
 **return** $[C(f) | \forall f \in \tilde{\mathcal{F}}_{eval}]$
$C \leftarrow \texttt{trainAttackClassifier}(\mathcal{D}_B, g_t)$
$m \leftarrow \texttt{fn}(x, y, g_t) : \texttt{isMember}([(x, y)], g_t, C)$

# 4. Evaluation

## 4.1. Evaluation Setup

### 4.1.1. Datasets

We use Librispeech [20] and TEDLIUM [21] in our experiments. To ensure that the inset and outset datapoints are distributed similarly and have the same set of speakers, we use only the training splits of the two datasets. We divide the Librispeech data into: $\text{LS}_{trn}^{tgt}$ (480 hours), $\text{LS}_{in}^{evl} \subset \text{LS}_{trn}^{tgt}$ (10 hours), $\text{LS}_{out}^{evl} \not\subset \text{LS}_{trn}^{tgt}$ (10 hours), $\text{LS}_{trn}^{att}$ (384 hours), and $\text{LS}_{out}^{att} \not\subset \text{LS}_{trn}^{att}$ (10 hours). Meanwhile, we divide the TEDLIUM data into: $\text{TED}_{trn}^{att}$ (338 hours), and $\text{TED}_{out}^{att} \not\subset \text{TED}_{trn}^{att}$ (8 hours). Alice uses $\text{LS}_{trn}^{tgt}$ to train $g_{target}$, while Bob wants to infer the membership of $\mathcal{D}_{attack} = \text{LS}_{in}^{evl} \cup \text{LS}_{out}^{evl}$. The rest of the datasets are used by Bob to train the proxy and attack models as detailed in 4.2.

### 4.1.2. Target and Proxy Model Details

The target and proxy models are RNN-Ts with different configurations. The target model, $g_{target}$, consists of a 6 layer LSTM [22] encoder, a 2 layer LSTM decoder and a Multi-Layered Perceptron (MLP) as the joint network. The decoder and encoder have 1024 units in each layer and output of the final layer is projected to 640 dimension before being passed to the joint network. The joint network creates a tensor, $J \in \mathbb{R}^{T \times U \times 640}$, such that $J_{tu} = h_t^{enc} + h_u^{dec}$, applies elementwise tanh to it and passes it to a MLP with one hidden layer containing 512 units. The proxy models, $g_{proxy}$, have the same architecture except that the encoder has 5 layers. Unless otherwise stated, all the models are trained with dropout [23] with $p = 0.3$, SpecAugment [24] settings from [25] and Adam optimzier [26], until the model converges or 40K iterations are completed. The learning rate starts at 1e-7 and warms up to 5e-4 over 1K iterations, stays constant for 20K iterations before decaying exponentially [27, 25] The batch size is set to 96 and 288 for the Librispeech and TEDLIUM models, respectively. The WER of the models on inset and outset data is presented in Table 1.

## 4.2. Attack Model Training

Bob follows Algorithm 1 for training proxy and attack models for each threat model. He sets $\mathcal{D}_B^1$ to $\text{TED}_{trn}^{att}$, $\text{LS}_{trn}^{att}$ and $\text{LS}_{trn}^{tgt}$ for the no knowledge (NK), partial knowledge (PK) and full knowledge (FK) threat models, respectively. For each of the threat models, he populates $\mathcal{D}_{attack}$ with features extracted from 10K utterances, of which 5K are sampled from $\mathcal{D}_B^1$ and 5K are sampled from either $\text{TED}_{out}^{att}$ in NK or $\text{LS}_{out}^{att}$ in PK and FK.

| Model | $\text{WER}_{\text{inset}}$ | $\text{WER}_{\text{outset}}$ | Attack Model Accuracy | | | |
|---|---|---|---|---|---|---|
| | | | RF | DT | LR | MLP64 |
| target | 8.7 | 14.8 | - | - | - | - |
| proxy-NK | 16.2 | 26.0 | 64.4 | 64.1 | 60.3 | 63.9 |
| proxy-PK | 8.5 | 17.0 | 67.8 | 67.1 | 62.9 | 67.4 |
| proxy-FK | 5.7 | 14.9 | 71.9 | 71.4 | 67.3 | 71.8 |

Table 1: *The WER of the target and proxy models on inset and outset data, and the accuracies of the corresponding attack models on data heldout from* $\mathcal{D}_{attack}$.

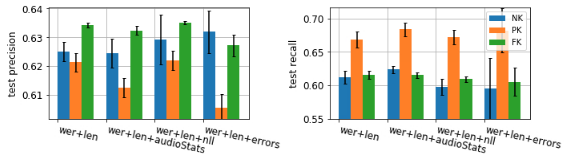



Figure 1: *Precision and recall of the RF attack model for different threat models and features computed from 4-best hypotheses.*

For each proxy model, Bob trains attack models with several types of binary classifiers, $C$, namely, Decision Tree (DT), Random Forest (RF), Logistic Regression (LR) and Multi-Layered Perceptron with one hidden layer containing 64 units (MLP64). The classification thresholds for LR and MLP64 are calibrated such that the difference between the true positive rate and false positive rate is maximized on the training set. Bob trains each type of classifier on the each of the four feature sets described in 3.2. The training process is repeated four times with different random seeds and different splits of $\mathcal{D}_{attack}$. Table 1 presents the accuracy of the classifiers trained with features from each proxy model, averaged across feature sets and data splits. While the differences were minute, the RF classifier was the most accurate so we discuss only its results in the subsequent sections.

## 4.3. Attack Results

Figure 1 shows the precision and recall achieved by the RF attack model against $g_{target}$, for different threat models and features computed from 4-best hypotheses. We observe that by using only *wer+lens*, Bob is able to achieve precision and recall more than 60% in all threat models, indicating the model's outputs do carry information about training data membership. Adding additional information like audio features, NLL and errors improves precision (slightly) in all cases, but adding NLL or errors decreases recall. This indicates that *wer+lens* alone provides a sufficiently strong signal for inferring membership. A more interesting observation is that the NK attacker closely tracks the precision and recall of the FK attacker indicating that *additional knowledge of the training data distribution is not necessary to mount an effective attack.* On the other hand the PK attacker suffers from high false positives, as indicated by the high recall and low precision scores. At first this seems counter intuitive, but by looking at Table 1 we note that for the PK proxy the WER for outset and inset data is lower, and their ratio is higher compared to the NK proxy, which indicates that the PK proxy has fit its training more closely than the NK proxy. Because of this the feature values extracted from the PK proxy may be too specific to its training data and may not generalize to $\mathcal{D}_{test}$ as well as the features from the NK proxy.

## 4.4. Analysis

To determine what causes the MIA to succeed, we analyse the no knowledge RF attack model trained on *wer+lens+nll*.

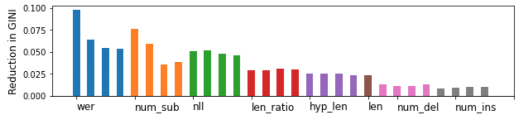



Figure 2: *Reduction in GINI impurity caused by each feature. The 4 bars for each feature correspond to the 4-best hypotheses.*

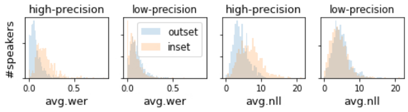



Figure 3: *Histogram of WER and NLL for inset and outset utterances of speakers with high and low precision scores*

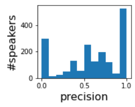



Figure 4: *Histogram of per-speaker MIA precision*

#### 4.4.1. Model Analysis

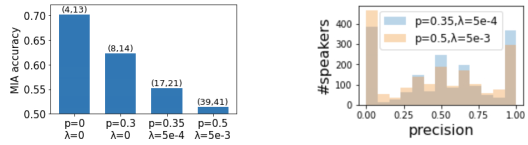
To determine Bob's primary attack vector we identify the features that are most "important" for predicting membership, i.e. cause the greatest reduction in GINI impurity [28, 29]. Figure 2 shows that WER is the most important feature, which means that $g_{target}$'s generalization gap is the primary source of vulnerability. Similar to the observation in [3] for machine translation models, we find that NLL is not very important, which suggests that the model does not always make overconfident predictions for inset data. This is validated by Figure 3, which shows that the distributions of NLL for inset and outset data is wider, and have greater overlap than the distributions of the WER.

These observations highlight two important differences in the nature of MIA attacks on sequence prediction tasks and classification tasks. First, in classification tasks the correctness of the model's prediction is binary and provides insufficient information for mounting a successful attack. Whereas in sequence prediction the predictions can be *partially* correct and can, thus, provide the attacker the additional information needed to mount a successful attack. Second, unlike a classification models, sequence prediction models use search methods, like beam search, that may reduce the influence of overconfidently predicted symbols by selecting lower probability symbols at some points if it increases the overall likelihood of the sequence. This would explain why NLL is not an important feature.

#### 4.4.2. Speaker-Level Analysis

To see if Bob is better at inferring membership of utterances from some speakers than others, we measure his precision on the utterances of each speaker and then plot the histogram of the values in Figure 4. To ensure that the values are meaningful, we consider speakers who have at least 1 inset and outset utterance. We find that for 25.6% of the speakers Bob's precision is more than 90% and for about 39% of the speakers it is greater than 75%, which means that Bob can precisely infer membership for more than a third of the speakers in this dataset.

To determine why certain speakers are more vulnerable we compared the average WER and NLL for speakers on which the attack model achieved high precision ($\geq$ 75%) with the speaker that yielded low precision ($<$ 75%). Figure 3 shows that for the speakers that yield high precision, the distributions of WER and NLL scores for inset samples and outset differ, with the outset distribution translated to the right. Whereas for low precision speakers the distributions are almost identical. This suggests that the model has overfit to some speakers and not others. Upon further investigation we discovered that speakers with high-precision contributed two more utterances on average to the training data, which suggests that over-representation in the training set *may* be linked to MIA vulnerability. However, further investigation is required to ascertain the extent to which this and other dataset sampling choices influence MIA vulnerability.




(a) (b)

Figure 5: *(a) MIA accuracy against regularized models with different Dropout probabilities (p) and L2 penalties ($\lambda$) with ($WER_{inset}$,$WER_{outset}$) on top of the bars. (b) Histogram of per-speaker MIA precision against two strongly regularized models.*

#### 4.4.3. Impact of Regularization

The above analysis echos the conclusion of [1], that overfitting is a sufficient condition for the success of MIA. To observe the impact of overfitting, we measure Bob's accuracy on attack models with different generalization gaps. To vary the generalization gap we train attack models with different dropout probabilities and L2 weight regualrization coefficients. The inset and outset WER for these models, and Bob's accuracy against them is presented in Figure 5a. We note that Bob's accuracy decreases with the generalization gap until it is very close to chance. However, by this point the model's WER is so high that it offer limited utility. Furthermore, Figure 5b reveals even under the strongest regularization Bob can still predict membership with more than 75% precision for 25.3% of the speakers, which is a (unacceptably) large population. These results suggest that *conventional regularization is not an effective defense against membership inference attacks*. If training data and computation power are in abundance, a better strategy may be to use differentially private training [30, 31], which would obscure peculiarities in each user's utterances and explicitly limit their influence on the model. Due to the lack of literature on the matter, training a differentially private ASR model with low WER may not be a trivial task, therefore we leave it to future work.

## 5. Conclusion

We have evaluated the vulnerability of RNN-T ASR models trained on public datasets to MIA under a black box threat model, using the shadow model technique proposed in [1]. We have found that the success of MIA is largely due to the model's generalization gap i.e. difference in the model's transcription accuracy on training and non-training data. We have also found that the generalization gap is non-uniformly distributed across speakers in the dataset which allows an attacker to infer the membership of utterances from certain speakers more accurately than others. To the best of our knowledge, no prior work has performed such a speaker-level analysis of MIA vulnerability. Based on this analysis, we recommend that future studies should report speaker-level MIA accuracy along with overall MIA accuracy. Finally, we observed that reducing the generalization gap using regularization reduces the overall accuracy of the MIA, however, the membership of utterances from a significant proportion of speakers can be still be inferred with high precision. These results indicate that investigating why the some speakers are more vulnerable to MIA and developing techniques to better defend against MIA would be promising directions for future work.

# 6. References


[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

[2] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[3] S. Hisamoto, M. Post, and K. Duh, "Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 49–63, 2020.

[4] 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 internet of things voice services," *arXiv preprint arXiv:1905.07082*, 2019.

[5] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," 2020.

[6] eMarketer, "Us voice assistant users 2019," https://www.emarketer.com/content/us-voice-assistant-users-2019, 2019.

[7] Microsoft, "Voice report: From answers to action: customer adoption of voice technology and digital assistants," https://advertiseonbing-blob.azureedge.net/blob/bingads/media/insight/whitepapers/2019/04%20apr/voice-report/bingads_2019_voicereport.pdf, 2019.

[8] C. A. C. Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," *arXiv preprint arXiv:2007.14321*, 2020.

[9] 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.

[10] 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," *arXiv preprint arXiv:1806.01246*, 2018.

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

[12] 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*, 2019, pp. 241–257.

[13] 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*, 2019, pp. 196–206.

[14] 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*, 2018, pp. 634–646.

[15] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 259–274.

[16] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[17] S. Punjabi, H. Arsikere, Z. Raeesy, C. Chandak, N. Bhave, A. Bansal, M. Müller, S. Murillo, A. Rastrow, S. Garimella *et al.*, "Streaming end-to-end bilingual asr systems with joint language identification," *arXiv preprint arXiv:2007.03900*, 2020.

[18] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.

[19] S. Seyfarth and P. Zhao, "Evaluating an automatic speech recognition service," https://aws.amazon.com/blogs/machine-learning/evaluating-an-automatic-speech-recognition-service/.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[21] A. Rousseau, P. Deléglise, and Y. Esteve, "Ted-lium: an automatic speech recognition dedicated corpus."

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[25] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Specaugment on large scale datasets," 2019.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[27] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," 2018.

[28] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification And Regression Trees*, 10 2017.

[29] K. Weinberger, "Decision trees," https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote17.html.

[30] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." 2014.

[31] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.