K-Aster: A novel Membership Inference Attack via Prediction Sensitivity

Ruoyi Li [1,*], Xinlong Zhao [2,*], Deshun Li [2] and Yuyin Tan [1,(✉)]

[1] School of Cyberspace Security, Hainan University, Haikou, Hainan,570228, China
[2] School of Computer Science and Technology, Hainan University, Haikou, Hainan,570228, China
990674@hainanu.edu.cn

Abstract. Membership Inference Attacks (MIA) are considered the fundamental privacy risk in Machine Learning (ML), which attempt to determine whether a specific data sample is training data for a target model. However, the recently proposed Aster only reports precision and recall for the member class without reporting false alarm rate (FAR) for the non-member class and the performance of target models. Additionally, Aster with Jacobi matrices requires the target model to output a vector of prediction probabilities, which can be easily defended when the model outputs only labels. In this paper, we propose a novel MIA method K-Aster, which only needs the output labels and partial training data of the target model to determine whether the data samples were used to train a given ML model. We obtain different output labels of the target model by data enhancement. Then we extract features from the labels to fit a line and quantify the prediction sensitivity with slope. Finally, we regard the samples with lower sensitivity as training data. Experimental results of attacks on Automatic Speech Recognition (ASR) systems show that our method is an important extension to Aster, which can achieve low FAR and high attack precision under non-classification tasks.

Keywords: Machine Learning, Membership Inference Attack, Aster, Prediction Sensitivity.

1 Introduction

Machine Learning (ML) has developed rapidly for the past decade with enormous advances in Big Data processing and computing power [1]. ML systems have been widely deployed in different application scenarios, e.g., machine translation [6], speech recognition [8], face recognition [9], and autonomous driving [10]. However, despite the convenience ML brings to the lives of people, it also brings serious challenges to data security and privacy. Training ML models requires a large amount of data, which is expensive to collect, organize, clean and label. Thus, model trainers may search for

Corresponding Author: Yuyin Tan. Email: 990674@hainanu.edu.cn

[*] Ruoyi Li and Xinlong Zhao—These authors contributed equally to this work.

The source code is available at https://github.com/13053676954/K-Aster.

alternative data sources, which may not always be legal. Therefore, the issue of whether personal data can be collected illegally and used to train ML models without their authorization is an urgent one. Membership inference attacks (MIA) have been proposed to counter threats in ML privacy [2].

The target of MIA is to determine whether a data sample belongs to the training set of a given model. MIA can audit private data for illegal training of models, e.g., using the results of membership inferences as legal evidence. MIAs may also seriously threaten data security, such as exposing the medical [22] and banking information of users. Most MIAs require a priori knowledge, and some of them require the structure and parameters of the target model [2, 4, 5, 7]. In addition, some MIAs require access to the training process of the model [3] and statistical information about the distribution of the training data [11]. However, most ML models are deployed as services with black-box access. We can only send input samples to the target model and receive the corresponding output.

Aster based on prediction sensitivity enables attacks in black-box scenarios [21]. The key idea of Aster is that training data from a fully trained ML model usually has a lower prediction sensitivity compared to test data. Low sensitivity means that when perturbing the feature values of the training samples, the predictions of the perturbed samples obtained from target models tend to be consistent with the original predictions. Aster perturbs the target samples and the black-box prediction interface of the target model. Non-training sample has a higher sensitivity to models in which it was not involved in training. Thus, with the difference in prediction sensitivity between training data and data seen by the model for the first time, Aster can execute attacks.

Despite its apparent success, Aster only reports precision and recall for the member class without reporting FAR for the non-member class and the performance of the target model (e.g., whether it is well-trained or whether there is overfitting). This report is often misleading. FAR shows the rate at which the attack model mislabeled non-training samples (non-members) as training samples (members). Most samples in reality belong to non-training sets, and most target models are well-trained. High FAR and overfitted target models make the membership reasoning task impractical. Meanwhile, the predictive sensitivity based on the Jacobi matrix requires the predictive probability vector of the target model rather than the output labels, and thus is not suitable for non-classification tasks. Aster can be easily defended if the model only exposes the predicted labels, i.e., the final model decision.

In this paper, based on the prediction sensitivity, we propose a novel MIA method K-Aster. K-Aster requires only the output labels of the target model and partial training data to determine whether a sample belongs to the training set of a given ML model. We employed data enhancement techniques such as modifying speech speed, cropping, and adding noise to generate additional samples, and then queried the target model with all the samples. We extract features from the output labels and then quantify the prediction sensitivity with the slope of the line fitted by these features. The sensitivity can reflect the relationship between the perturbation of each feature and the amplitude of the corresponding label change. We train the attack model using the predicted sensitivity and its corresponding membership labels, which will infer the membership of given samples. In this paper, we experiment with automatic speech recognition (ASR) systems. During the experiments, we report the FAR of non-member classes and the performance of the target model. The experimental results show that our method is a significant extension to Aster, which enables low FAR and high attack precision under non-classification tasks.

Our major contributions are summarized as follows:

1. We report the performance of Aster in non-member classes. Experiments show that Aster has a high FAR and is impractical.
2. Based on the prediction sensitivity, we propose a novel MIA K-Aster. Compared to Aster, we achieve the attack using only the predicted labels of target models without predicting probability vectors.
3. We attack ASR systems and report both FAR and performance of the target model. In a more realistic scenario, experiments show that K-Aster achieves low FAR and high attack precision.

The rest of the paper is organized as follows. Section 2 describes related work on MIA and Aster. Section 3 describes the attack method of K-Aster. Section 4 describes the experimental setup, the performance of Aster, and the experimental results of K-Aster. Section 5 summarizes our work.

2 Related work

In this section, we describe related research work in MIA and Aster.

2.1 Membership Inference Attack

Shokri et al. [2] first proposed MIA for ML models, which is intended to infer whether a given data sample belongs to the training set of the target model. When there is only black-box access to the target model, they construct a set of shadow models with known training and non-training samples to imitate the target model. Then they use the outputs of the shadow models to compose an attack dataset and train a set of attack models. This leakage of membership information may lead the attacker to infer certain private information about the data samples, which is usually used to measure whether the target model has privacy security concerns. In addition to using shadow models, [3, 4, 5, 14, 15] use other information from the target model to execute MIA. Salem et al. [12] argue that MIA can be achieved without shadow models by setting a threshold for the predictive confidence of the model.

Choquette-Choo et al. [16] proposed the first label-only MIA. They achieve MIA by evaluating the robustness of the model prediction labels under perturbation. Li et al. [17] proposed two types of label-only MIA called transfer attacks and boundary attacks. The principle of the transfer attack is that the attacker has a dataset with the same distribution as the training set of the target model. Then they construct shadow models to achieve MIA locally. The principle of the boundary attack is that the attacker adds noise to the target sample and attempts to change the predictive labels of the target model. By measuring the amount of noise added, the attacker can determine whether the target sample is used to train the target model.

Recently, Hyun Kwon et al. [13] proposed a selective MIA. By using the proposed method, membership or non-membership can be inferred by generating a decision model from the prediction of the inference models and training the confidence values for the data corresponding to the selected class. Martin Bertran et al. [19] proposed a scalable MIA via quantile regression. The attack is based on performing quantile regression on the distribution of confidence scores induced by the model under attack on points that are not used in training. Shi Chen et al. [20] proposed High Precision MIA (HP-MIA), a novel two-stage attack scheme that leverages membership exclusion techniques to guarantee high membership prediction precision.

2.2 Aster

Lan Liu et al. [21] proposed an MIA Aster based on prediction sensitivity. The target of Aster is to disclose the privacy of the training data of target models without knowledge of model and the training data. Aster only needs the black-box API of the target model and a data sample to determine whether that sample is used to train a given ML model.

Aster is based on the observation that ML models are less sensitive to perturbations in feature values on training samples compared to non-training samples. As the training process progresses, ML models become more confident in their predictions of the training data. When the training process is complete, the trained model is highly robust to the training data. Thus, for a fully trained ML model, perturbations to the training samples do not lead to significant changes in the prediction output of the model. Non-trained samples will have a higher sensitivity to models that are not involved.

Aster executes the attack with the difference in prediction sensitivity. Specifically, Aster uses the Jacobi matrix to capture the predictions of a given sample for a target model. The Jacobi matrix consists of the relationship between the feature values of the input samples and the output predictions of the target model. Compared to non-training samples, the Jacobian matrix of training samples has a smaller paradigm. Therefore, Aster uses the paradigm of the Jacobi matrix to measure the prediction sensitivity and infer which samples are likely to be training data.

3 Methodology of K-Aster

In this section, we describe the specific attack methods of K-Aster.

3.1 Overall process of K-Aster

Given data samples and a target model that outputs only predictive labels, the target of K-Aster is to infer whether the samples are used to train the model. The process is shown in **Fig. 1**. K-Aster first inputs the data samples into the target model for prediction with data enhancement. Then we extract the features of the labels and fit a line to obtain the prediction sensitivity. Finally, K-Aster inferences on the target samples based on the sensitivity to determine whether the samples are from the training set of the target model. Thus, the method can be divided into three stages: extraction of labeled features, quantification of prediction sensitivity and membership inference.

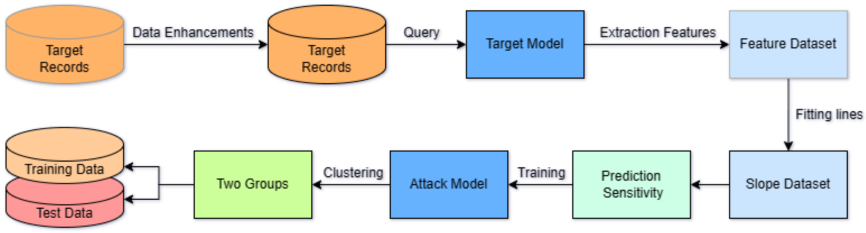


Fig. 1. Overview of K-Aster.

3.2 Extract features of labels

With the limitation that the target model only outputs predictive labels, we can only extract features by inputting data samples and getting predictive labels. To measure the difference between training and non-training samples, we use the following categories of features:

Error rate (ER): we use the ratio of the number of incorrect words to the total number of words in the model output text as the error rate. Specifically, we quantify the difference between the predicted text and the target text using the number of replacement, deletion, and insertion words in the output text as a percentage of the total.

Similarity (Sim): We adopt the similarity between the target text and the predicted text as another metric. Specifically, we use cosine similarity and Euclidean distance methods to calculate the similarity between real text and predicted text.

To obtain more features, we used data enhancement techniques on speech samples, including changing the speech speed, cropping, and adding five different types of noise. Then we extracted features of error rate and similarity from the data samples after each data enhancement.

3.3 Quantitative prediction sensitivity

We extracted three types of features including error rate, cosine similarity and similarity calculated using the Euclidean distance method and related features after seven different types of data enhancement. We obtain the slope of each category of features by fitting a line. Based on the observation that the slopes of the lines fitted by the features of the training samples are usually smaller than that of the non-training samples. To strengthen our observations and prove the feasibility of K-Aster, we experimented on two datasets. The results are completely consistent with our motivation. As shown in **Table 1**, the average slope of the samples from the training set is smaller than the average slope of the samples from the test set. Thus, from preliminary experiments, we can see that the samples in the model training set are less sensitive to perturbations. The prediction sensitivity can be captured by the slope of the line fitted by the sample features.

Table 1. Mean slope (prediction sensitivity).

Experimental setup		Training set			Test set		
Datasets	Models	ER	Sim1	Sim2	ER	Sim1	Sim2
LibriSpeech	ASR 1	1.329	-0.050	5.492	1.881	-0.096	7.839
CommonVoice(en)	ASR 2	1.059	-0.062	24.076	1.521	-0.090	26.409

3.4 Membership inference

Combining the prediction sensitivity with labels representing member states (member or non-member) forms the attack dataset and trains the binary classifier as an attack model in a supervised manner. Different from Aster, we can execute the attack on a training dataset with only one sample. In the attack process, we query the target model with a specific data sample and obtain the output predicted text. We extract the features of the output using the same method and then send it to the attack model to determine whether the sample belongs to the training set of the target model. The output of the attack model is 0, indicating that the sample to be detected does not belong to the training set of the target model. The output of 1 indicates that it is a training set sample.

4 Experimental Evidence

In this section, we first describe the basic setup of the experiment, and then we evaluate the performance of Aster and K-Aster.

4.1 Experiment Setup

Priori Knowledge. In the experiments with K-Aster, we consider a more practical setup. We can only obtain partial information about the training data and the predictive labels by accessing the target model. This assumption limits Aster in the sense that we do not have access to the vector of predicted probabilities for a given input. We do not have access to any relevant information about the target model, such as structure, type, parameters, training algorithms and setup. This assumption is feasible because in reality, there are different methods by which we can determine that some public dataset is in the training set of the target model.

Datasets. Similar to the experimental setup of Aster, we use two public datasets, UCI Adult and MNIST, and randomly select 10,000 samples from them to form the training set for the Aster target model. To evaluate the performance of K-Aster, we use the LibriSpeech and Common Voice(en) datasets to train the target model.The LibriSpeech dataset is a large database containing approximately 1,000 hours of English speech read aloud. The Common Voice(en) dataset is an audio dataset containing 9283 hours of recorded audio, including 7335 hours of verified speech, covering 60 languages.

Target Models. To evaluate the performance of Aster on the non-member class, we use Logistic Regression (LR), Random Forest (RF) and Support Vector Machine (SVM) as the target models. And to evaluate the performance of K-Aster, we used SpeechBrain [18] to construct ASR system as the target model and implement speech data enhancement.

Automatic Speech Recognition (ASR) system is a technology that converts speech into text. This technology has a wide range of applications in many fields, such as customer service, voice assistants, assistive devices for the hearing impaired, etc. ASR systems work in three main steps: feature extraction, acoustic modeling, and language modeling. In the feature extraction stage, the system extracts useful features from the input speech signal. Then, the acoustic model uses these features to predict possible phonemes or words. Finally, the language model will select the most likely word sequences based on context and grammar rules.

We trained the ASR model 1 on the LibriSpeech dataset. Model 1 uses a combined block of Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Deep Neural Network (DNN) as the acoustic model and an RNN structure as the speech model. The model has an average error rate of about 2.37% on the training set and 3.09% on the test set.

We trained the ASR model Model2 on the Common Voice(en) dataset using the Transformer architecture as acoustic model and language model.The model had an average error rate of about 7.73% on the training set and 15.69% on the test set.

The training and testing gaps of our ASR target model are within reasonable range, there is no overfitting, and closer to realistic scenarios. In contrast, in some previous works, the generalization gap is sometimes greater than 35% [2, 23], 50% [5], or 80% [24], and their target models are not realistic.

Attack Models. In the experiments for evaluating the performance of Aster, we use the same spectral clustering algorithm as in [21] to construct the attack model. And in the experiments to evaluate the performance of K-Aster, we use the three models, Decision Tree (DT), Logistic Regression (LR) and Random Forest (RF), as the attack models.

Evaluation Metrics. In our experiments, we use performance metrics including accuracy, precision, recall, F1-score, and FAR. Accuracy is the percentage of total samples that are predicted correctly. Precision is the probability that all samples predicted to be members of the training set are actually members of the training set. Recall is the probability that one of the samples that is actually a member of the training set is predicted to be a member of the training set. F1-score is the balance between precision (i.e., attack precision) and recall (i.e., attack coverage).FAR is the probability that an attack model incorrectly labels a non-training sample as a training sample.

4.2 Evaluation of Aster

In our experiments, we execute Aster on three ML models and two datasets. We measure the performance of Aster more comprehensively by inferring five different numbers of data samples. The results of our experiments are summarized in **Table 2**. The average attack accuracy of Aster across all target models is 0.540, the average attack precision is 0.543, the average attack recall is 0.760, the average attack F1-score is 0.605, and the average attack FAR is 0.680.

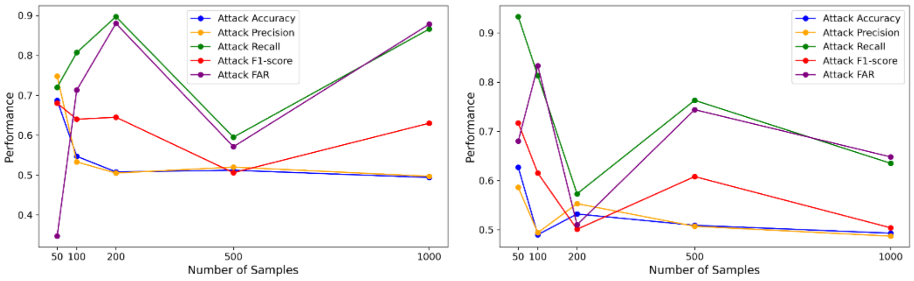
Specifically, we first discuss the performance of the Aster attacks on the LR model. Aster has an mean attack accuracy of 0.532, a mean attack precision of 0.557, a mean attack recall of 0.669, a mean attack F1-score of 0.551, and a mean attack FAR of 0.604. Then we discuss the performance of the Aster attacks on the RF model. Aster has a mean attack accuracy of 0.565, a mean attack precision of 0.556, a mean attack recall of 0.831, a mean attack F1-score of 0.660, and a mean attack FAR of 0.700. Finally, we discuss the effect of Aster on the SVM model. Aster has a mean attack accuracy of 0.522, a mean attack precision of 0.517, a mean attack recall of 0.781, a mean attack F1-score of 0.603, and a mean attack FAR of 0.737.

Table 2. Attack Performance of Aster.

Experimental setup			Metrics		
Model-Samples-Dataset	Accuracy	Precision	Recall	F1-score	FAR
DT-50-Adult	0.580	0.543	1.000	0.704	**0.840**
LR-50-Adult	0.660	0.900	0.360	0.514	0.040
RF-50-Adult	0.860	0.821	0.920	0.868	0.200
SVM-50-Adult	0.540	0.524	0.880	0.657	**0.800**
DT-100-Adult	0.470	0.482	0.800	0.602	**0.860**
LR-100-Adult	0.510	0.506	0.840	0.632	**0.820**
RF-100-Adult	0.590	0.569	0.740	0.643	0.560
SVM-100-Adult	0.540	0.525	0.840	0.646	**0.760**
DT-200-Adult	0.505	0.503	0.960	0.660	**0.950**
LR-200-Adult	0.520	0.512	0.860	0.642	**0.820**
RF-200-Adult	0.490	0.495	0.970	0.655	**0.990**
SVM-200-Adult	0.515	0.509	0.860	0.639	**0.830**
DT-500-Adult	0.492	0.495	0.720	0.586	**0.736**
LR-500-Adult	0.492	0.495	0.812	0.615	**0.828**
RF-500-Adult	0.528	0.519	0.784	0.624	**0.728**
SVM-500-Adult	0.516	0.547	0.188	0.280	0.156
DT-1000-Adult	0.526	0.523	0.586	0.553	0.534
LR-1000-Adult	0.501	0.501	0.830	0.625	**0.828**
RF-1000-Adult	0.494	0.497	0.972	0.658	**0.984**
SVM-1000-Adult	0.487	0.492	0.796	0.608	**0.822**

Experimental setup			Metrics		
Model-Samples-Dataset	Accuracy	Precision	Recall	F1-score	FAR
DT-50-Mnist	0.440	0.000	0.000	nan	0.120
LR-50-Mnist	0.560	0.533	0.960	0.686	**0.840**
RF-50-Mnist	0.720	0.667	0.880	0.759	0.440
SVM-50-Mnist	0.600	0.558	0.960	0.706	**0.760**
DT-100-Mnist	0.490	0.480	0.240	0.320	0.260
LR-100-Mnist	0.550	0.530	0.880	0.662	**0.780**
RF-100-Mnist	0.430	0.457	0.740	0.565	**0.880**
SVM-100-Mnist	0.490	0.494	0.820	0.617	**0.840**
DT-200-Mnist	0.570	0.557	0.680	0.613	0.540
LR-200-Mnist	0.530	0.615	0.160	0.254	0.100
RF-200-Mnist	0.525	0.519	0.700	0.596	**0.650**
SVM-200-Mnist	0.540	0.524	0.860	0.652	**0.780**
DT-500-Mnist	0.506	0.506	0.508	0.507	0.496
LR-500-Mnist	0.512	0.507	0.816	0.626	**0.792**
RF-500-Mnist	0.522	0.517	0.676	0.586	**0.632**
SVM-500-Mnist	0.494	0.496	0.796	0.611	**0.808**
DT-1000-Mnist	0.497	0.495	0.314	0.384	0.320
LR-1000-Mnist	0.488	0.467	0.170	0.249	0.194
RF-1000-Mnist	0.495	0.497	0.924	0.647	**0.934**
SVM-1000-Mnist	0.497	0.498	0.810	0.617	**0.816**

The experimental results show that the Aster attacks are more successful in terms of precision and FAR only for the combination of LR model and Adult dataset, the combination of RF model and Adult dataset, and the combination of RF model and Mnist dataset. And all the above three good attack results exist in the case of small data samples. The attacks of Aster on LR model and Adult dataset suffer from low Recall. In most of the attacks, Recall is always higher than Precision and FAR is still high, which suggests that there are many misleading predictions and MI attacks may not be meaningful. In addition, we observed the trend of the effect of five different numbers of data samples under the Aster attack. The results are shown in **Fig. 2**. The results show that the precision of the Aster attack is decreasing in general as the amount of sample data increases. Aster performs poorly with larger amounts of data.



(a) Adult Dataset (b) Mnist Dataset

Fig. 2. The impacts of the number of samples.

4.3 Performance of K-Aster

We execute K-Aster on two ASR models and two speech datasets to evaluate its performance. Our experimental results are summarized in **Table 3**. For the ASR system, K-Aster has a mean attack accuracy of 0.698, a mean attack precision of 0.678, a mean attack recall of 0.755, a mean attack F1-score of 0.713, and a mean attack FAR of 0.368.

Specifically, we first discuss the performance of the K-Aster attack on the LibriSpeech dataset. K-Aster has a mean attack accuracy of 0.656, a mean attack precision of 0.646, a mean attack recall of 0.781, a mean attack F1-score of 0.707, and a mean attack FAR of 0.486. Then we discuss the performance of the K-Aster attack on the Common Voice (en) dataset. K-Aster has a mean attack accuracy of 0.741, a mean attack precision of 0.709, and a mean attack FAR of 0.486. Experimental results show that our method is a significant extension to Aster, achieving low FAR and high attack precision under non-classification tasks.

Table 3. Attack Performance of K-Aster.

Experimental setup	Metrics				
Attack Model-Dataset	Accuracy	Precision	Recall	F1-score	FAR
DT-LibriSpeech	0.660	0.655	0.763	0.705	0.457
LR-LibriSpeech	0.638	0.628	0.780	0.696	0.523
RF-LibriSpeech	0.669	0.655	0.800	0.720	0.479
DT-CommonVoice(en)	0.743	0.715	0.723	0.719	0.241
LR-CommonVoice(en)	0.721	0.694	0.692	0.693	0.256
RF-CommonVoice(en)	0.758	0.719	0.772	0.744	0.253

5 Conclusion

With the rapid development of ML, MIA has been widely studied as a form of privacy leakage for ML models. However, recently proposed Aster based on prediction sensitivity suffers from high FAR and is easily defended in realistic scenarios. In this paper, we propose a novel MIA K-Aster, which requires only the output labels of the target model and partial training data to determine whether a sample is used to train a given ML model. We extract a series of features from the output labels of the target model and quantify the prediction sensitivity with the slope of the line fitted by these features. It reflects the relationship between the perturbation of each feature and the sharpness of the change in the corresponding label. Experimental results of attacks on ASR systems show that our method is an important extension to Aster, achieving low FAR and high attack precision under non-classification tasks. We expect our work to reveal personal data privacy issues that are more relevant to realistic scenarios and to contribute to the development of defenses against MIAs with prediction sensitivity.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (62162021), the Hainan Provincial Natural Science Foundation of China (623RC448).

References

1. G. Liu, T. Xu, R. Zhang, Z. Wang, C. Wang and L. Liu, "Gradient-Leaks: Enabling Black-Box Membership Inference Attacks Against Machine Learning Models," in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 427-440, 2024.
2. R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 3-18.
3. Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2017). Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. 2018 IEEE 31st Computer Security Foundations Symposium (CSF), 268-282.
4. B. Wu et al., “Characterizing membership privacy in stochastic gradientlangevin dynamics,” Proc. AAAI Conf. Artif. Intell., vol. 34, no. 04, pp. 6372–6379, 2020.
5. 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," 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019, pp. 739-753.
6. Wongyung Nam and Beakcheol Jang. 2024. A survey on multimodal bidirectional machine learning translation of image and natural language processing. Expert Syst. Appl. 235, C (Jan 2024).
7. A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jegou, “White-box vs black-box: Bayes optimal strategies for membership inference,” in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 5558–5567.
8. Douglas O'Shaughnessy. 2024. Trends and developments in automatic speech recognition research. Comput. Speech Lang. 83, C (Jan 2024).
9. Yulan Guo, Hanyun Wang, Longguang Wang, Yinjie Lei, Li Liu, and Mohammed Bennamoun. 2023. 3D Face Recognition: Two Decades of Progress and Prospects. ACM Comput. Surv. 56, 3, Article 54 (March 2024), 39 pages.

10. X. Shi, Y. D. Wong, C. Chai, M. Z. -F. Li, T. Chen and Z. Zeng, "Automatic Clustering for Unsupervised Risk Diagnosis of Vehicle Driving for Smart Road," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 17451-17465, Oct. 2022.
11. 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," in IEEE Transactions on Computational Social Systems, vol. 6, no. 5, pp. 907-921, Oct. 2019.
12. Jia, J., Salem, A., Backes, M., Zhang, Y., & Gong, N.Z. (2019). MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security.
13. H. Yan et al., "Membership Inference Attacks Against Deep Learning Models via Logits Distribution," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 3799-3808, 1 Sept.-Oct. 2023.
14. Klas Leino and Matt Fredrikson. 2020. Stolen memories: leveraging model memorization for calibrated white-box membership inference. In Proceedings of the 29th USENIX Conference on Security Symposium (SEC'20). USENIX Association, USA, Article 91, 1605–1622.
15. B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, “Practical blind membership inference attack via differential comparisons,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2021, pp. 1–17.
16. C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, “Label-only membership inference attacks,” in Proc. 38th Int. Conf. Mach. Learn., 2021, pp. 1964–1974.
17. Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21). Association for Computing Machinery, New York, NY, USA, 880–895.
18. Ravanelli M , Parcollet T , Plantinga P , et al. SpeechBrain: A General-Purpose Speech Toolkit[J]. 2021.
19. Bertran, Martin & Tang, Shuai & Kearns, Michael & Morgenstern, Jamie & Roth, Aaron & Wu, Zhiwei. (2023). Scalable Membership Inference Attacks via Quantile Regression.
20. Shi Chen, Wennan Wang, Yubin Zhong, Zuobin Ying, Weixuan Tang, Zijie Pan, HP-MIA: A novel membership inference attack scheme for high membership prediction precision, Computers & Security, Volume 136, 2024.
21. L. Liu, Y. Wang, G. Liu, K. Peng and C. Wang, "Membership Inference Attacks Against Machine Learning Models via Prediction Sensitivity," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 2341-2347, 1 May-June 2023.
22. Ben Hamida, S., Ben Hamida, S., Snoun, A. et al. The influence of dropout and residual connection against membership inference attacks on transformer model: a neuro generative disease case study. Multimed Tools Appl 83, 16231–16253 (2024).
23. Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. Towards measuring membership privacy. arXiv preprint arXiv:1712.09136, 2017.
24. Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In Proceedings of the 28th USENIX Conference on Security Symposium (SEC'19). USENIX Association, USA, 1895–1912.