AdversariaLeak: External Information Leakage Attack Using Adversarial Samples on Face Recognition Systems

Roye Katzav[1], Amit Giloni[1], Edita Grolman[1], Hiroo Saito[2], Tomoyuki Shibata[2], Tsukasa Omino[2], Misaki Komatsu[2], Yoshikazu Hanatani[2], Yuval Elovici[1], and Asaf Shabtai[1]

[1] Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev
{royek, hacmona, edita}@post.bgu.ac.il; {elovici, shabtaia}@bgu.ac.il
[2] Corporate Research & Development Center, Toshiba Corporation
{hiroo5.saito, tomoyuki1.shibata, tsukasa.omino, misaki1.komatsu, yoshikazu.hanatani}@toshiba.co.jp

Abstract. Face recognition (FR) systems are vulnerable to external information leakage (EIL) attacks, which can reveal sensitive information about the training data, thus compromising the confidentiality of the company's proprietary and the privacy of the individuals concerned. Existing EIL attacks mainly rely on unrealistic assumptions, such as a high query budget for the attacker and massive computational power, resulting in impractical EIL attacks. We present AdversariaLeak, a novel and practical query-based EIL attack that targets the face verification model of the FR systems by using carefully selected adversarial samples. AdversariaLeak uses substitute models to craft adversarial samples, which are then handpicked to infer sensitive information. Our extensive evaluation on the MAAD-Face and CelebA datasets, which includes over 200 different target models, shows that AdversariaLeak outperforms state-of-the-art EIL attacks in inferring the property that best characterizes the FR model's training set while maintaining a small query budget and practical attacker assumptions.

Keywords: Face Recognition · External Information Leakage · Adversarial Samples

1 Introduction

Face recognition (FR) systems are deployed in various domains, including physical access control [22], surveillance [18], and mobile device security [13]. The core component of modern FR systems is a large-scale deep neural network (DNN) that performs face verification, *i.e.*, identifying whether two images belong to the same individual [1, 23]. The inclusion of DNNs in the recognition pipeline causes the system to be vulnerable to various security breaches and attacks [11, 26]. Those attacks are major concerns since an attacker can perform an external

information leakage (EIL) attack to infer implicit statistical properties about the target model's training set. This would result in a violation of the confidentiality of the company's proprietary (specifically the model's training set) and harm the privacy of the included individuals [24,30]. These properties could include sensitive and personal information, such as gender, ethnicity, age and geolocation. Existing works have demonstrated that different machine learning (ML) models, including face verification (FV) models, are vulnerable to EIL attacks [6, 10, 19, 30]. However, these works often rely on unrealistic assumptions about the attacker's knowledge (complete knowledge of the model's architecture and parameters, access to the training set collection space, etc.) and computational resources (training hundreds of substitute models, access to many data samples, etc.) [17, 24]. As a result, the current state-of-the-art EIL attacks may not be practical in a real-world attack scenario.

We present AdversariaLeak, a novel and practical EIL attack that targets the FV model of the FR system and assumes a realistic attacker knowledge while using a small number of queries to the target model. Given an attacker's sample set, the attacker trains a substitute model for each property value and uses the models to craft adversarial samples, i.e., samples that mislead the model and cause it to make an incorrect prediction. A few of the crafted adversarial samples, referred to as the unique adversarial samples, are then handpicked and used to query the target model, maliciously leading it to reveal its training set properties. By using the unique adversarial samples to model the substitute models' differences, the attacker can examine the effect of the property distribution in the training set on the model's decision boundary.

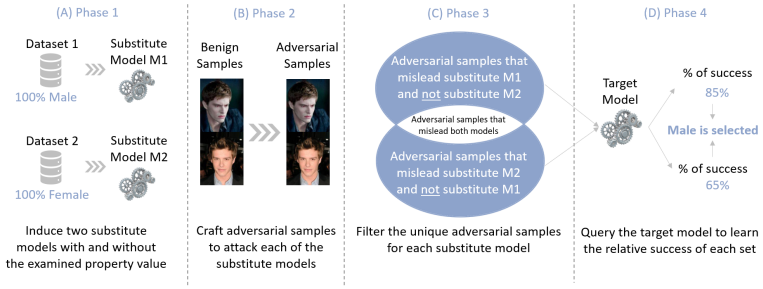

Fig. 1: An overview of AdversariaLeak with the 'gender' property; (A) induce substitute models; (B) craft adversarial samples for each of the substitute models; (C) filter the unique adversarial samples; (D) query the target model to infer the property value.

We evaluate AdversariaLeak in two attack scenarios and 180 different target models using the CelebA dataset [15] and three of its implicit properties: the individuals' age group, gender, and the presence of a 5 o'clock shadow (a day-old beard). Additionally, to show that AdversariaLeak can be generalized to other datasets, we extended our evaluation to an additional dataset, the MAAD-

Face [25] dataset, and tested the gender property (male). AdversariaLeak was evaluated on five different property distributions: 100%, 75%, 50%, 25%, and 0%, each of which represents the proportion of the positive property value in the training set. If the examined property is gender, then a distribution of 25% means that 25% of the identities in the training set are male and the rest are female. Our evaluation demonstrates that AdversariaLeak outperforms existing EIL attacks by: *i)* successfully inferring the property value that best characterizes the target model's training set; *ii)* holding practical and realistic attacker assumptions; and *iii)* using a small query budget that includes only 24 to 60 queries to the target model (depending on the attack scenario).

Understanding and mitigating the information leakage that AdversariaLeak exposes is essential to: 1) Ensure fair and secure deployment of widely used FR systems. 2) Safeguard personal data. 3) Maintain the integrity of these FR systems that rely heavily on sensitive biometric authentication. 4) Highlight potential biases and weaknesses, prompting the development of more robust and fair FR systems. The main contributions of this paper are:

- The first EIL attack that successfully utilizes adversarial samples.
- The first EIL attack that assumes realistic attacker restrictions, which is done by using a small number of substitute models and a minimum number of queries.
- A generic attack that is not dependent on specific properties or any prior knowledge about the training set.

2 Related Work

EIL attacks can target different types of models, such as fully connected neural networks (FCNNs) [10], convolutional neural networks (CNNs) [24], collaborative learning models [19], and generative adversarial networks (GANs) [30]. Depending on the attacker's knowledge regarding the target model's architecture and parameters, existing EIL attacks can be categorized as complete-knowledge (white-box), no-knowledge (black-box), and partial-knowledge (gray-box) attacks. The first EIL attack, which was a white-box attack, trained substitute models on datasets with different property distributions and used the models' parameters to train a meta-classifier for predicting the external property value [2]. This attack was extended to FCNNs [10] and CNNs [24].

Although white-box attacks can be effective, they are considered impractical due to their access assumptions. Black-box and gray-box attacks have been proposed as more practical alternatives. Existing black-box attacks rely on querying the target model and examining the predictions according to different property values, such as examining the difference in accuracy between samples belonging to different property values [24] and setting a performance threshold that distinguishes between property values [24]. Gray-box attacks also rely on querying the target model, yet require additional partial knowledge about the target model. In particular, the existing gray-box EIL attacks assume access to the training set collection space [6,17]. In both attacks, the attacker inserts poisoned samples

into the training set to increase the correlation between the examined property and a specific label learned by the model. Then, the attacker trains substitute models on different property distributions using the poisoned samples and trains a meta-classifier on the substitute models' confidence scores. Finally, the attacker uses the target model's confidence scores to infer its property value. The two attacks differ in the meta-classifier's training data - Mahloujifar *et al.* used the raw confidence scores and Chaudhari *et al.* used normalized confidence scores.

Most of the EIL attacks assume impractical attacker assumptions: *i)* white-box attacks assume complete access to the target model; *ii)* gray-box attacks require access to the training data collection space or access to inject new samples into the training set; and *iii)* the black-box threshold test attack [24] requires training over 100 substitute models, leaving the loss test attack [24] as the only practical EIL attack. In our evaluation, we compared AdversariaLeak to the loss test and the threshold test attacks under realistic assumptions - the threshold test using two substitute models and the loss test in its original form.

AdversariaLeak utilizes adversarial evasion attacks to perform an EIL attack. Although it is the first to do so, several works utilized adversarial attacks for privacy-related tasks, such as a membership inference attack [7], a model extraction attack [29], and defending against attribute inference attacks [14]. For additional background and related work see supplementary material.

3 AdversariaLeak

In this section, we introduce AdversariaLeak's components and structure (illustrated in Fig. 1). Given an external dataset and access to query the target ML model, an attacker can use AdversariaLeak to infer the property value that best characterizes the target model's training data, i.e., the property distribution which is the closest to the target model's training set distribution. AdversariaLeak is composed of four main phases: *1)* the attacker uses the external dataset to train substitute models, each of which is trained on a different extreme property distribution, resulting in one substitute model per property value; *2)* the attacker leverages an adversarial evasion attack to craft adversarial samples for each of the substitute models; *3)* for each substitute model, the attacker selects unique adversarial samples that mislead solely that model; and *4)* the attacker queries the target model with the unique samples and infers the property value according to the model's behavior. By using unique adversarial samples that mislead only a specific substitute model, the attacker obtains a unique representation of each substitute model's behavior. Since the substitute models only differ in the property distribution of their training set, that representation can be considered unique to the property distribution characterizing the training set.

The notation used is as follows: Let $S_T \sim D_{ST}$ be the set of face image pairs used to train target model T, which was derived from identity distribution D_{ST}. Let P be the examined property and $\{v_0, v_1, ..., v_n, v_T\} \in P_{values}$ be P's possible values, where v_T is the value that characterizes most of the samples in S_T. Let $S_A \sim D_{SA}$ be a set of face image pairs collected by the attacker and derived

from identity distribution D_{SA}. Note that there is a clear separation between the identities in D_{ST} and D_{SA}, *i.e.*, the datasets S_A and S_T are mutually exclusive.

3.1 Phase 1 - Attacker's Substitute Models

In this phase, the attacker uses the external dataset S_A to train several substitute models, each of which corresponds to a different property value. First, the attacker forms several data subsets from S_A, where each subset $S_{A0} \cup S_{A1} \cup ... \cup S_{An} \subset S_A$ corresponds to a different property value. For example, in the case in which P is "gender", the attacker would form two subsets $S_{A_{male}} \cup S_{A_{female}} \subset S_A$, where $S_{A_{male}}$ only includes pairs comprised of male identities, and $S_{A_{female}}$ only includes pairs comprised of female identities. Each subset $S_{Ai} \subset S_A$ is then used to train a substitute model M_{Ai}. In the example mentioned above in which P is "gender", the attacker uses $S_{A_{male}}$ to train $M_{A_{male}}$ and $S_{A_{female}}$ to train $M_{A_{female}}$. As a result, the attacker obtains several substitute models, each modulating the target model's expected behavior in the case it was trained on an extreme property value distribution (0% and 100%), i.e., the decision boundary of each substitute model is an estimation of the target model's decision boundary in the case it was trained on an extreme property value distribution.

3.2 Phase 2 - Craft Adversarial Samples

In this phase, the attacker crafts adversarial samples for each substitute model M_{Ai}. First, the attacker selects pairs of images that belong to the same identity (labeled as "the same person") and were correctly labeled by M_{Ai}. This dataset, denoted as $S_{A_{adv}}$, is selected from the samples that remained in S_A after performing phase 1, *i.e.*, $S_{A_{adv}} = S_A \setminus \bigcup_{i=1}^{n} S_{Ai}$, where $\setminus$ is the operation of sets subtraction. Then, the attacker uses the samples of $S_{A_{adv}}$ to craft a set of adversarial samples for each substitute model M_{Ai}, *i.e.*, samples that mislead model M_{Ai} to believe that the images in the pair belong to two different identities (predict "not the same person"). The adversarial sample set for each model M_{Ai} is crafted by applying the optimization process used in existing adversarial evasion attacks on $S_{A_{adv}}$. During this optimization, an adversarial noise is optimized so that when it is added to the pair's pixels, they remain as close to the original values as possible while causing M_{Ai} to output the wrong prediction ("not the same person"). This objective is optimized by minimizing a cost function. The general adversarial optimization process is presented in equation 1:

$$\widehat{pair_i} = pair_i + \varepsilon opt(\nabla_{pair_i} L(pair_i, y)) \tag{1}$$

where $pair_i \in S_{A_{adv}}$ is an image pair, *opt* is the optimization function, and $\nabla_{pair_i} L(pair_i, y)$ are the derivatives of cost function L with respect to $pair_i$ and the true prediction y. The adversarial attack used by AdversariaLeak is empirically selected. The adversarial sample set crafted for model M_{Ai} is denoted as $\overline{S}_{Ai_{adv}}$. By crafting a set of adversarial samples for each substitute model,

the attacker can obtain a more comprehensive representation of the behavior of each extreme property distribution. The crafted adversarial samples rely on the gap between the true decision boundary and the model's decision boundary, i.e., the difference between the truth and the model's prediction. Since each substitute model M_{Ai} is trained on a different training set (with a different property distribution), those gaps would differ between the different substitute models, i.e., different adversarial samples would be crafted. In contrast, when considering a different model (denoted as M_j) trained on the same distribution as M_{Ai}, the two models would most likely have similar decision boundary gaps, i.e., some of the adversarial samples crafted for M_{Ai} would mislead M_j as well.

3.3 Phase 3 - Select Unique Samples

In this phase, the attacker selects the unique adversarial samples from each adversarial set $\overline{S}_{Ai_{adv}}$ obtained in phase 2, i.e., the adversarial samples crafted for M_{Ai} that mislead only M_{Ai} and not mislead other substitute models. To obtain the unique adversarial samples for each substitute model M_{Ai}, the attacker first handpicks $\overline{S}_{Ai_{adv}}+$, which is the subset of $\overline{S}_{Ai_{adv}}$ that succeeds in misleading M_{Ai}. Then, the attacker selects the unique samples from $\overline{S}_{Ai_{adv}}+$ by performing the process presented in equation 2:

$$\overline{S}_{Ai_{uq}} = \{x \in \overline{S}_{Ai_{adv}} + |\forall j \neq i, M_{Ai}(x) \neq M_{Aj}(x)\} \tag{2}$$

where x is an adversarial sample from the $\overline{S}_{Ai_{adv}}+$ set, $M_{Ai}(x)$ is the prediction of M_{Ai} for sample x (the incorrect prediction), and $M_{Aj}(x)$ is the prediction of a different substitute model that is not M_{Ai}. By filtering the unique adversarial samples of each substitute model, the attacker emphasizes the behavioral differences between the different substitute models. The handpicked unique adversarial samples mislead M_{Ai} by using the decision boundary gaps that are unique for M_{Ai}, i.e., unique for the property distribution that M_{Ai} was trained on. Therefore, these samples are expected to cause models that were trained on a similar distribution as M_{Ai} to incorrectly classify the samples as "not the same person", and other models to correctly classify them as "the same person".

3.4 Phase 4 - Inferring the Property Value

In this phase, the attacker queries the target model with the unique samples and infers the property value. First, the attacker queries the target model T with each of the unique samples sets $\overline{S}_{Ai_{uq}}$. Then, the attacker aggregates T's predictions into a single score for each $\overline{S}_{Ai_{uq}}$. The aggregated score is the fraction of adversarial samples that mislead the target model (denoted as FMS, which is the fraction of misleading samples) and is presented in equation 3:

$$FMS(\overline{S}_{Ai_{uq}}) = \frac{|\{x \in \overline{S}_{Ai_{uq}} | T(x) = 0\}|}{|\overline{S}_{Ai_{uq}}|} \tag{3}$$

where $x \in \overline{S}_{Ai_{uq}}$ is a unique adversarial sample crafted for model M_{Ai}, $T(x)$ is the target model's prediction for x, and 0 denotes the class "not the same person". By examining the FMS scores, the attacker can determine which $\overline{S}_{Ai_{uq}}$ was the most successful in misleading target model T, i.e., which substitute model M_{Ai} has the decision boundary most similar to T. The property value that characterizes the training set of the substitute model with the highest FMS score is chosen by the attacker as the property value that best characterizes the training set of the target model T.

4 Evaluation

All the experiments were performed using three cross-validations, with 15, 32, and 42 as the random seeds. In total, AdversariaLeak was evaluated in 420 different experiments: 1) 360 different experiments on the CelebA dataset [15]: 3 cross-validations * 2 backbones * 3 properties * 5 distributions * 2 evasion attacks * 2 attack scenarios; and 2) 60 different experiments on the MAAD-Face dataset [25]: 3 cross-validations * 2 backbones * 1 property * 5 distributions * 2 evasion attacks * 1 attack scenario. Additional evaluation details, as well as AdversariaLeak's code, can be found in the supplementary material.

4.1 Datasets

To properly evaluate AdversariaLeak, the datasets used should have: *i)* a large number of identities; *ii)* a large number of images per identity; and *iii)* implicit property annotations. We used the CelebA [15] and MAAD-Face [25] datasets, which fulfill these requirements. Note, other datasets were explored but did not meet the research requirements (see supplementary material). The CelebA dataset [15] (based on celebrity images) consists of 202,599 images of 10,177 identities, with annotations for 40 properties, including age, gender, and the presence of a 5 o'clock shadow (a day-old beard). The MAAD-Face dataset [25] is an enhanced version of the VGGFace2 dataset [4] (based on Google Image Search), utilizing different property annotations for each image, including the gender property. To ensure enough images per identity to create pairs of "the same person", we removed the dataset's identities which had only one image.

The datasets were divided into three separate sets: *i)* 50% of the identities were used as the target model's training sets; *ii)* 25% of the identities were used as the attack evaluation set, which was used to train and evaluate the substitute models; and *iii)* 25% of the identities were used as the attack crafting sample set, which was used to craft the adversarial samples used by AdversariaLeak. Note, we ensured that identities in the target model's dataset will not appear in the attacker datasets. Therefore, these datasets are not of the same distribution.

Since the target model's task is face verification, the images in all the sets were paired and labeled, where "the same person" is denoted as one, and "not the same person" is denoted as zero. The attack evaluation set and crafting sample set are both balanced with respect to the examined property.

To perform the evaluation on target models with different property distributions (100%, 75%, 50%, 25%, and 0%), only part of the target model's training set was used in each evaluation. To create a training set with a specific property distribution, the following steps were performed: (1) all the samples of the identities from the least common value were selected, i.e., those identities that fall under a property value that has the lowest occurrence compared to other property values (*e.g.*, if the property is gender and there are 500 males and 300 females identities, then, female is the least common value). (2) a portion of the identities from the other property value is added based on the property distribution required.

4.2 Experimental Settings

Evaluation Environment. All experiments were performed on the CentOS Linux 7 (Core) operating system using 60 GB of memory and an NVIDIA GeForce RTX 3090 GPU card. In addition, the experimental code was written using Python 3.9.15, Sklearn 1.0.2, NumPy 1.23.5, adversarial-robustness-toolbox (ART) 1.12.2 and PyTorch 1.13.0.

Attack scenarios. We examined AdversariaLeak in two different attack scenarios. The first (scenario 1) is to examine AdversariaLeak when the head's architecture is identical for both the target and substitute models. It was evaluated only on the CelebA dataset [15] due to its less likely attacker knowledge. The second (scenario 2) is to examine AdversariaLeak under stricter assumptions, in which the head's architecture used for the target and substitute models is different. Since this attack scenario is more likely, it was evaluated using both the CelebA [15] and MAAD-Face [25] datasets.

FR Models. The target and substitute models used are composed of a backbone (IResNet100[3] or RepVGG B0 [27]) and a head, as face verification models typically are; additional information on the models and backbones can be found in the supplementary material. Throughout the experiments, the heads, used for both types of models, receive the subtraction (common approach in FR) of the two embedding vectors, one for each face image, and output the probability that the two images are of the same person. Two different head architectures were used: *1)* A head consists of two dense layers with 64 and 8 neurons respectively, and the rectified linear unit (ReLU) activation function. *2)* A head consists of seven dense layers with 512, 256, 128, 64, 32, 16 and 8 neurons respectively, and the rectified linear unit (ReLU) activation function. The final layer in both heads consists of a single neuron and the sigmoid activation function. In attack scenario 1 the first architecture was used whereas in attack scenario 2 the target model used the second architecture and the attacker used the first. The weights and biases for each layer were randomly initialized, and each head was trained using the Adam optimizer with a batch size of 64 and a learning rate of 0.0001. For CelebA [15], the first head was trained for 10 epochs; for MAAD-Face [25], for 20 epochs; and the second head for 30 epochs in both datasets. When using

[3] https://sota.nizhib.ai/pytorch-insightface/iresnet100-73e07ba7.pth

the CelebA dataset [15], we used pre-trained backbones, i.e., we trained only the head. However, when using the MAAD-Face dataset [25], we fine-tuned the backbones' weights to reach sufficient performance. This fine-tuning required an additional 10 epochs, training both the head and backbone together.

Adversarial attacks. We examined AdversariaLeak by using two different adversarial evasion attacks to craft the adversarial samples - Carlini-Wagner [5] and the Projected Gradient Descent (PGD) attack [16] as implemented in the ART framework [20]. For Additional specifications see supplementary material.

Tested Properties. We evaluated AdversariaLeak using three properties: (1) 5 o'clock shadow - whether an identity has a day-old beard or not; (2) young - whether an identity is young or not; and (3) male - whether an identity is male or not. 'Male' and 'Young' are commonly used properties in FR and EIL domains; while '5 o'clock shadow' is more challenging due to its delicate physical aspects. Note, 5 o'clock shadow and young properties were evaluated in the CelebA dataset [15] evaluation; and male property was evaluated in both the CelebA [15] and MAAD-Face [25] datasets evaluation. In the evaluation, five different proportions of the positive property value were evaluated - 0%, 25%, 50%, 75%, and 100%. As all examined properties are binary, AdversariaLeak was applied with two substitute models, i.e., one for each extreme distribution (100% and 0%).

Compared Attacks. We compared AdversariaLeak to two state-of-the-art EIL attacks - the loss test and threshold test attacks [24]. In the loss test attack, we create two attacker sets with 0% and 100% property distributions. Then, we query the target model with each set, evaluate its performance (using accuracy), and select the property values of the set that achieved better performance. In the threshold test attack, we used two substitute models with extreme property distributions 0% and 100%, i.e., we adjusted this attack to realistic attacker assumptions. Then, we create two data sets S_0 and S_1 with 0% and 100% property distributions respectively, and identify which of them maximizes the performance gap between the substitute models (denoted as S_k). Then, we find the threshold λ that maximizes the ability to distinguish between models that were trained on S_k and all others. Finally, we queried the target model with S_k and select S_k's property value if the performance received is higher or equal to λ.

5 Experimental Results

The results were obtained using RepVGG_B0 [8] and IResNet100 [3] backbones; IResNet100 results are in the supplementary material due to space limits.

Attack Results All of the results presented were validated by three cross-validations. Fig. 2 presents the results for AdversariaLeak on the CelebA dataset [15] using the Carlini-Wagner (CW) [5] (plots (a)-(c) and (g)-(i)) and the Projected Gradient Decent adversarial attack (PGD) [16] (plots (d)-(f) and (j)-(l)) with a query budget of 3,000 samples at most. Each plot in Fig. 2 presents the results obtained for the 5 o'clock shadow (blue bars), denoted as 5OCS; young

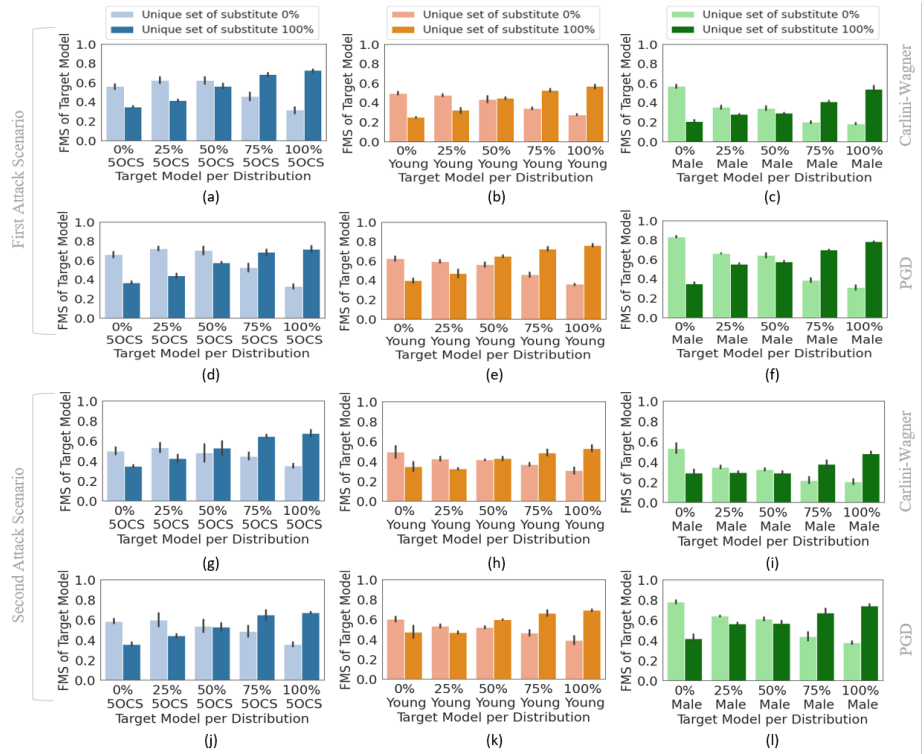

Fig. 2: Results of AdversariaLeak on the CelebA dataset [15].

(orange bars); and male (green bars) properties. The results are on attack scenario 1 (plots (a)-(f)) and attack scenario 2 (plots (g)-(l)). Each plot presents the fraction of adversarial samples that mislead the target model, denoted as FMS (y-axis), for each target model trained on the five different property distributions (x-axis). For each target model, the FMS metric is presented for the unique samples crafted for the two substitute models of AdversariaLeak: *i)* 0% property value distribution (the lighter bar); and *ii)* 100% property value distribution (the darker bar). According to AdversariaLeak, when the FMS of the 0% substitute model (the lighter bar) is higher, the property value that best characterizes the target model training set is the "0" value (*e.g.*, not 5 o'clock shadow, not young, not male) and vice versa. Similarly, Fig. 3 presents the results for AdversariaLeak on the MAAD-Face dataset [25] using the Carlini-Wagner (CW) [5] (plot (a)) and the Projected Gradient Decent adversarial attack (PGD) [16] (plot (b)) with a query budget of 3,000 samples at most.

Overall, it can be seen that AdversariaLeak succeeds in identifying the property value that best characterizes the target model's training set regardless of the adversarial attack used, the attack scenario and the dataset, i.e., when the target model is of the 0% and 25% distributions, the 0% substitute model has a

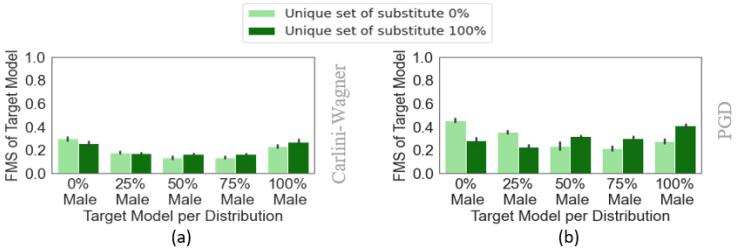

Fig. 3: Results of AdversariaLeak on the MAAD-Face dataset [25].

Table 1: Results of AdversariaLeak (Our), loss test (LTA) and threshold test (TDTA) attacks [24], on the CelebA [15] and MAAD-Face [25] datasets.

Property	Scenario	Target Model Per Distribution														
		0%			25%			50%			75%			100%		
		LTA	TDTA	Our	LTA	TDTA	Our	LTA	TDTA	Our	LTA	TDTA	Our	LTA	TDTA	Our
5OCS (CelebA [15])	1	Not 5OCS	5OCS	Not 5OCS	Not 5OCS	5OCS	Not 5OCS	Not 5OCS	5OCS	Not 5OCS	5OCS	5OCS	5OCS	5OCS	5OCS	5OCS
	2	Not 5OCS	5OCS	Not 5OCS	Not 5OCS	5OCS	Not 5OCS	Not 5OCS	5OCS	5OCS	5OCS	5OCS	5OCS	5OCS	5OCS	5OCS
Young (CelebA [15])	1	Not Young	Not Young	Not Young	Not Young	Not Young	Not Young	Not Young	Not Young	Young	Not Young	Not Young	Young	Young	Not Young	Young
	2	Not Young	Not Young	Not Young	Not Young	Not Young	Not Young	Not Young	Not Young	Young	Not Young	Not Young	Young	Not Young	Not Young	Young
Male (CelebA [15])	1	Not Male	Not Male	Not Male	Male	Not Male	Not Male	Male	Not Male	Not Male	Male	Male	Male	Male	Male	Male
	2	Not Male	Not Male	Not Male	Male	Male	Not Male	Male	Male	Not Male	Male	Male	Male	Male	Male	Male
Male (MAAD-Face [25])	2	Not Male	Male	Not Male	Not Male	Male	Not Male	Male	Male	Male	Male	Male	Male	Male	Male	Male

higher FMS, and when the target model is of the 100% and 75% distributions, the 100% substitute model has a higher FMS. This success is statistically significant (see supplementary material). In addition, when examining the results on the target model trained on a balanced distribution (the 50% distribution), it can be seen that most of the gaps between the FMS of the two substitute models (lighter and darker bars) are relatively smaller than those in other distributions (Section 6). Furthermore, the standard deviation (the black line in each bar) is less than 0.09 in most of the experiments, indicating that AdversariaLeak's results are consistent. Moreover, we note that the standard deviation slightly increased from scenario 1 to scenario 2 in Fig. 2 (which is consistent with the scenarios' difficulty) with no noticeable changes in the FMS gaps.

Query Budget Effect Fig. 4 presents the results for AdversariaLeak on the CelebA dataset [15] using the Carlini-Wagner [5] (plots (a)-(e) and (k)-(o)) and the PGD [16] (plots (f)-(j) and (p)-(t)) attacks. The results are on attack scenario 1 (plots (a)-(j)) and attack scenario 2 (plots (k)-(t)). Each plot in Fig. 4 presents the gap (y-axis) between the FMS of samples crafted on a substitute model trained on 0% of the property and on a substitute model trained on 100% of the property (i.e., $FMS_{0\%} - FMS_{100\%}$), across different query budgets (x-axis), which ranges between $[1, 200]$, and different properties (colors) for each target model trained on the five different property distributions (denoted as PD). Note, the query budget represents the number of unique samples from both the substitute models, i.e., $query\ budget = unique\_set\_0\% + unique\_set\_100\%$. The percentage gap, presented on the y-axis, can reflect AdversariaLeak's final decision. When negative, the attacker infers that the target model training set

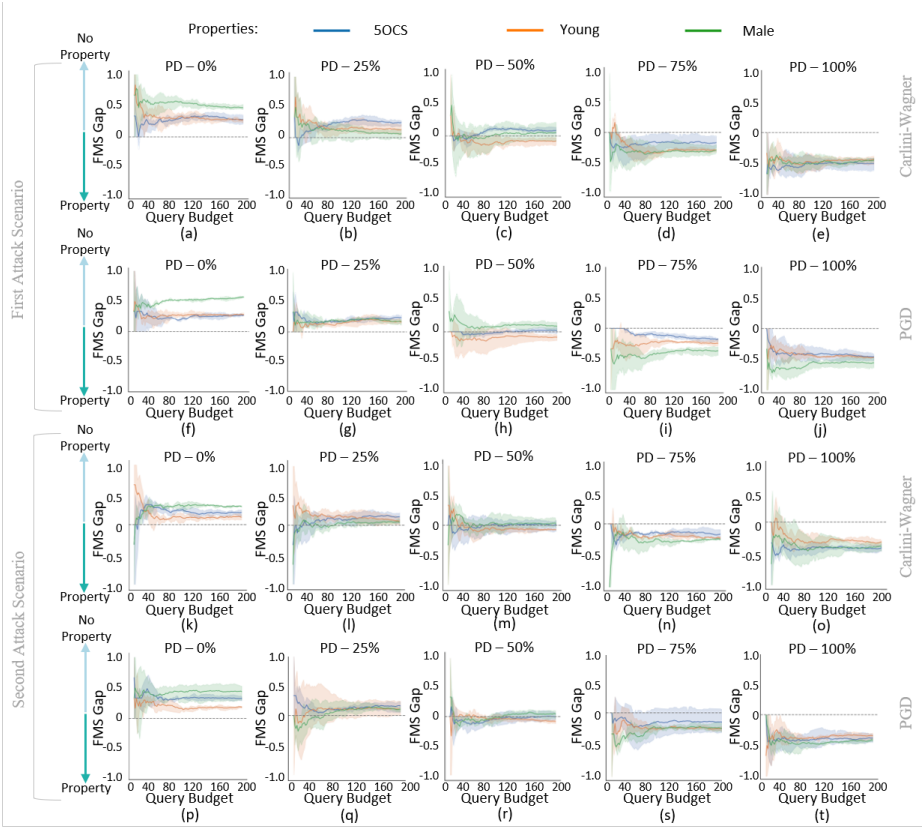

Fig. 4: Query budget results of AdversariaLeak on the CelebA dataset [15].

mainly consists of the examined property (*e.g.*, 5 o'clock shadow, young, male), and when positive, the attacker infers the opposite (*e.g.*, not 5 o'clock shadow, not young, not male). To control the query budget used, the number of unique adversarial samples taken from the unique sample sets to query the target model was limited. This was done by selecting the unique samples that received the highest confidence score from the substitute model they were crafted on, i.e., the samples that mislead the substitute model the most. Similarly, Fig. 5 presents the results for AdversariaLeak on the MAAD-Face dataset [25] using the Carlini-Wagner [5] (plots (a)-(e)) and the PGD [16] (plots (f)-(j)) attacks. Overall, it can be seen that AdversariaLeak succeeds in identifying the property value that best characterizes the target model's training set, regardless of the adversarial attack used, the evaluated scenario and the dataset, by using a minimum of 24 queries to the target model in scenario 1 and 60 in scenario 2. This indicates that AdversariaLeak outperforms existing attacks and reduces the query number required to perform a successful EIL attack.

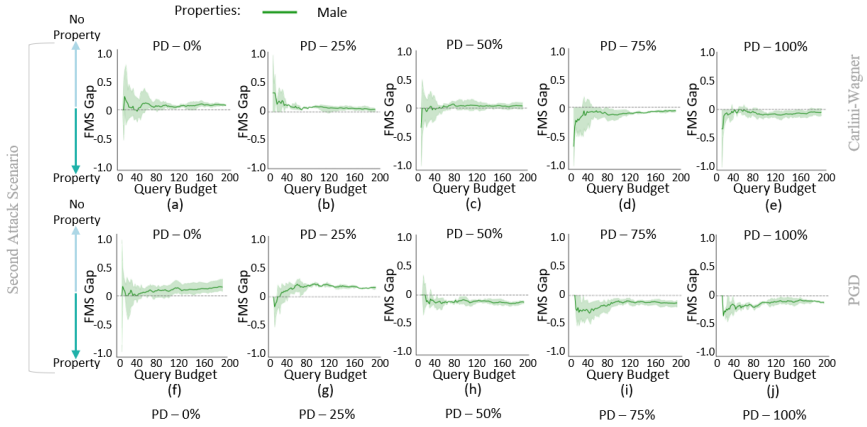

Fig. 5: Query budget results of AdversariaLeak on the MAAD-Face dataset [25].

Comparison to EIL Attacks Table 1 presents the results of the loss test (LTA) and threshold test (TDTA) attacks [24] and AdversariaLeak's final results on the CelebA [15] and MAAD-Face [25] datasets. The presented LTA and TDTA results were obtained by using the entire attack crafting sample set in their benign form, i.e., before performing the adversarial sample crafting process. Note, that the attack crafting samples set is not part of the training process of the models and is used only to query the models in any of the compared methods. The attack crafting samples set of the CelebA dataset [15] contained about $7,088$, $14,000$, and $23,405$ images with a standard deviation of 209.08, $6,922.65$, and $14,454.51$ in the 5 o'clock shadow, young, and male properties experiments respectively. The attack crafting samples set of the MAAD-Face dataset [25] contained about $656,680$ images with a standard deviation of $3,413.99$ in the male property's experiments.

From Table 1 we can see that in the experiments on the 5 o'clock shadow property, TDTA failed in half of the experiments (the grey cells), whereas LTA and AdversariaLeak achieved similar results, i.e., the final property value chosen by the attacker is identical. In the experiments on the young and male properties, AdversariaLeak outperformed LTA and TDTA, i.e., AdversariaLeak succeeded in inferring the correct property value in all the examined experiments whereas LTA and TDTA did not (the grey cells). Moreover, when examining both scenarios, we can see that the LTA is less stable when the attack scenario is stricter (scenario 2). Furthermore, while the LTA and TDTA failed to infer the correct property value in some of the experiments when using tens of thousands of queries to the target model ,for example, using maximum query budget of 24,000 samples in each experiment (for a complete query budget analysis see supplementary material), AdversariaLeak succeeded in inferring the correct property value in all of the experiments, with a minimal number of 24 to 60 queries.

6 Discussion

Throughout the experiments, interesting attack behavior was discovered in the experiments related to the target models trained on a 50% property distribution. In those experiments, the gap between the FMS of the substitute model trained on 0% of the property and the substitute model trained on 100% of the property (i.e., $FMS_{0\%} - FMS_{100\%}$) should be close or equal to zero (inconclusive decision). However, in most of those experiments on the CelebA dataset [15], AdversariaLeak showed a clear preference for specific property values, i.e., not 5 o'clock shadow, young, and not male values. These tendencies to favor specific property values can be explained by examining the only component that is not affected by the property distribution in all target models trained on the CelebA dataset [15], i.e., the pre-trained backbone that was used. The RepVGG_B0 backbone was trained on a clean version of the MS-Celeb-1M dataset [12], which varied with respect to the identities' age (there is no clear indication for a specific age distribution), yet it contains mainly female identities, which causes it to have more identities without a beard. It can be seen that the gender distribution (mostly female) and 5 o'clock shadow distribution (mostly without beard) of the MS-Celeb-1M dataset leaks into AdversariaLeak's results for the target models with a 50% property distribution when the backbone is pre-trained. Contrary in the MAAD-Face dataset [25] this phenomenon is not observed which can be explained by the fact that we have fine-tuned the backbone. Those observations indicate that the property distribution of the backbone's training set may affect the attack results (extended in the supplementary material).

To mitigate the risk posed by EIL attacks, several countermeasures can be considered: adversarial training [16], differential privacy [9], fine-tuning with diverse datasets [28] and robust architectures with defensive distillation [21]. These measures collectively improve model robustness and privacy, which could reduce the vulnerability to EIL attacks (extended in the supplementary material).

7 Conclusion and Future Work

In this paper, we presented AdversariaLeak, a novel EIL attack for FR systems that infers the property value that best characterizes the distribution of the target model training set. AdversariaLeak uses a minimal number of substitute models to craft and handpick a set of unique adversarial samples, which are used to query the target model. The experimental results demonstrate that, in contrast to existing attacks, AdversariaLeak is successful and suitable for real-world attack scenarios due to its use of a minimal query budget of 24 to 60 queries and practical attacker assumptions. The principles behind AdversariaLeak are not limited to FR systems. Future work may include applying AdversariaLeak in different domains such as tabular data domains; adjusting it to other computer vision tasks, such as object detection; and estimating the exact target model's distribution.

References

1. Arashloo, S.R., Kittler, J.: Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features. IEEE Transactions on Information Forensics and Security **9**(12), 2100–2109 (2014)
2. Ateniese, G., Mancini, L.V., Spognardi, A., Villani, A., Vitali, D., Felici, G.: Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. International Journal of Security and Networks **10**(3), 137–150 (2015)
3. Behrmann, J., Grathwohl, W., Chen, R.T., Duvenaud, D., Jacobsen, J.H.: Invertible residual networks. In: International Conference on Machine Learning. pp. 573–582. PMLR (2019)
4. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). pp. 67–74. IEEE (2018)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)
6. Chaudhari, H., Abascal, J., Oprea, A., Jagielski, M., Tramèr, F., Ullman, J.: Snap: Efficient extraction of private properties with poisoning. In: 2023 IEEE Symposium on Security and Privacy (SP). pp. 400–417. IEEE (2023)
7. Choquette-Choo, C.A., Tramer, F., Carlini, N., Papernot, N.: Label-only membership inference attacks. In: International conference on machine learning. pp. 1964–1974. PMLR (2021)
8. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13733–13742 (2021)
9. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
10. Ganju, K., Wang, Q., Yang, W., Gunter, C.A., Borisov, N.: Property inference attacks on fully connected neural networks using permutation invariant representations. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 619–633 (2018)
11. Goswami, G., Ratha, N., Agarwal, A., Singh, R., Vatsa, M.: Unravelling robustness of deep learning based face recognition against adversarial attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
12. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European conference on computer vision. pp. 87–102. Springer (2016)
13. Ijiri, Y., Sakuragi, M., Lao, S.: Security management for mobile devices by face recognition. In: 7th International Conference on Mobile Data Management (MDM'06). pp. 49–49. IEEE (2006)
14. Jia, J., Gong, N.Z.: Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. In: 27th {USENIX} security symposium ({USENIX} security 18). pp. 513–529 (2018)
15. Liu, Z., Luo, P., Wang, X., Tang, X.: Large-scale celebfaces attributes (celeba) dataset. Retrieved August **15**(2018), 11 (2018)
16. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)

17. Mahloujifar, S., Ghosh, E., Chase, M.: Property inference from poisoning. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1569–1569. IEEE Computer Society (2022)
18. Majeed, F., Khan, F.Z., Iqbal, M.J., Nazir, M.: Real-time surveillance system based on facial recognition using yolov5. In: 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC). pp. 1–6. IEEE (2021)
19. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 691–706. IEEE (2019)
20. Nicolae, M.I., Sinn, M., Tran, M.N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., et al.: Adversarial robustness toolbox v1. 0.0. arXiv preprint arXiv:1807.01069 (2018)
21. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE symposium on security and privacy (SP). pp. 582–597. IEEE (2016)
22. Ribeiro, R., Lopes, D., Neves, A.: Access control in the wild using face verification. In: Intelligent Video Surveillance, pp. 1–18. IntechOpen (2018)
23. Seo, H.J., Milanfar, P.: Face verification using the lark representation. IEEE Transactions on Information Forensics and Security **6**(4), 1275–1286 (2011)
24. Suri, A., Evans, D.: Formalizing and estimating distribution inference risks. Proceedings on Privacy Enhancing Technologies **2022** (2022)
25. Terhörst, P., Fährmann, D., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: Maad-face: A massively annotated attribute dataset for face images. IEEE Transactions on Information Forensics and Security **16**, 3942–3957 (2021)
26. Vakhshiteh, F., Nickabadi, A., Ramachandra, R.: Adversarial attacks against face recognition: A comprehensive study. IEEE Access **9**, 92735–92756 (2021)
27. Wang, J., Liu, Y., Hu, Y., Shi, H., Mei, T.: Facex-zoo: A pytorch toolbox for face recognition. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3779–3782 (2021)
28. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Advances in neural information processing systems **27** (2014)
29. Yu, H., Yang, K., Zhang, T., Tsai, Y.Y., Ho, T.Y., Jin, Y.: Cloudleak: Large-scale deep learning models stealing through adversarial examples. In: NDSS (2020)
30. Zhou, J., Chen, Y., Shen, C., Zhang, Y.: Property inference attacks against gans. In: 30th Network and Distributed System Security Symposium (NDSS 2022) (2022)