Least Privilege Learning for Attribute Obfuscation

Glen Brown(✉), Jesus Martinez-del-Rincon, and Paul Miller

Centre for Secure Information Technologies, Queen's University Belfast, Belfast, UK
{gbrown29,j.martinez-del-rincon,p.miller}@qub.ac.uk
https://www.qub.ac.uk/ecit/CSIT

Abstract. As machine learning systems become ever more prevalent in everyday life, the need to secure such systems is becoming a critically important area in cybersecurity research. In this work, we address the "feature misuse" attack vector, where the features output by a model are abused to perform a function that they were not originally designed for, such as determining a person's gender in a facial verification system. To mitigate this, we take the security concept of "least privilege", where a system can only access resources it explicitly needs to complete its task, and apply it to training deep neural networks. This "least privilege learning" ensures features do not contain information regarding protected attributes that are superfluous to the primary task, reducing the potential attack surface for feature misuse and reducing undesired information leakage. In this paper, we present two main contributions. Firstly, a novel training paradigm that enables least privilege learning by obfuscating protected attributes in verification and re-identification scenarios. Secondly, a comprehensive evaluation framework for models trained with least privilege learning, encompassing multiple datasets and three application settings: verification, re-identification, and attribute prediction.

Keywords: Least privilege learning · Attribute obfuscation · Machine learning feature misuse · Adversarial learning · Protecting deep learning models

1 Introduction

Security of machine learning is emerging as a new frontier for cybersecurity. Since Goodmann et al.'s seminal paper [15], adversarial learning has been an area of much research activity. When considering the security of machine learning, one needs to consider the confidentiality, integrity and availability challenges posed by each phase of the machine learning cycle. Of particular interest is the training phase, which is one of the most critical steps, since it establishes the baseline behaviour of the application. This is the area that is most likely to present unique security challenges, as learning is at the core of the machine learning process.

The training stage consists of running the model iteratively with a baseline data set for which the desired output is known. With each iteration, the

C. Wallraven et al. (Eds.): ACPR 2021, LNCS 13189, pp. 142–156, 2022.
https://doi.org/10.1007/978-3-031-02444-3_11

model parameters are adjusted to achieve more accurate performance, and this is repeated until an optimal or acceptable level of accuracy is achieved. It is critical that the training data set is of high quality, as any inaccuracies or inconsistencies can lead to the model behaving incorrectly. A typical example of a biometric access control system which utilises facial analysis involves a user with an identifying card. The card stores (or can be used to retrieve) a previously generated feature vector which encodes the properties of the user's face. On presentation of the card to the system, a camera takes a new image and encodes it as a new feature vector. If the distance between the new and stored feature vectors is below a pre-determined threshold, they are deemed to match and access is granted. Otherwise, access is denied.

However the performance of the training task should not be the only goal, since features generated by a model can be used, without further training, for inference purposes other than that which they were intended. For example, a biometric face recognition system developed for access control, may contain a model whose features can be used to recognise a person's gender, age or ethnic group. Facial biometric systems are becoming ever more prevalent, so potential avenues of abuse against such systems need to be investigated.

Feature misuse is the attack which we address in this paper. To do so we introduce the concept here of least privileged learning. Along with Need to Know, Least Privilege is one of the underlying principles of security which states that an entity should only be given access to a specific resource that is needed to perform a task. In the context of learning, we can apply least privilege to ensure that a model, or its features, can only be used for that purpose for which it was designed and nothing else. Hence, in the biometric access control example we want to ensure that the features learnt for verifying an identity cannot be used to determine a person's gender for example. To achieve this we propose the use of a second unlearning task in which the system is trained to become ignorant of the attribute of interest, in this case gender. To accomplish this, we construct a novel training paradigm that can obfuscate attributes in facial biometric systems, as well as a comprehensive framework for evaluating systems that utilise least privilege learning.

2 Related Work

Different training paradigms have been proposed to mitigate the encoding of unwanted attributes into models. One approach is to alter the input data. Authors in [12] use a style transfer system to remove features from the images which are correlated with demographic attributes, producing "neutral" faces. Authors in [6] also use a style transfer technique, but instead use a data augmentation procedure to increase the demographic diversity of the training data by transforming images into other demographics.

Another approach alters the training paradigm. [18] demonstrated an *Adversarial Debiasing* model on word embeddings using an adversarial training scenario [7], combining a *predictor* which learns the primary task, and an *adversary* which attempts to predict the protected attribute. The loss across the full model is formulated in such a way that updates made to the *predictor* are prevented from

decreasing the *adversary's* loss, thereby removing bias encoded in the *predictor's* output. Similarly, Authors in [13] use a *gradient reversal layer* in a multitask-training model based on [5] which updates the weights of the model in opposition to the secondary task, namely protected attribute prediction. Research in [1] introduced a Joint Learning and Unlearning (JLU) framework utilising a confusion loss (inspired by [17]) where they successfully trained a gender-blind age classifier. The confusion loss computes the cross-entropy between the model output and a uniform distribution, moving the model towards a state of randomness with respect to protected attribute prediction. While our work is related, we apply the general method in conjunction with metric learning and a different training paradigm to facial verification and/or re-identification tasks across several standard "in-the-wild" datasets, as opposed to the attribute discrimination tasks in the original paper.

3 Method

The basis of our least privilege learning framework takes inspiration from both Domain-Adversarial Neural Networks (DANN) [5] and Generative Adversarial Networks (GAN) [7], while using a confusion loss as in [1]. We start with a Multi-Task Learning model with the two tasks being *verification* and *attribute discrimination* (Sect. 3.1). Then, we utilise a penalising loss function, also known as a confusion loss (Sect. 3.3), and a two-stage training step (Sect. 3.2) to correctly back-propagate the penalised gradients, while still allowing effective learning. With this process, the model can learn a suitable representation of the latent space with which Verification/Re-identification is possible while not leaking information about the protected attribute in the resulting embedding vectors.

3.1 Multi-task Learning

The basic model architecture consists of a central CNN Backbone F which functions as a deep *feature extractor*, and a separate Multilayer Perceptron P called the *Attribute Prediction Branch (APB)* which takes the generated features x' as inputs and discriminates the value of the protected attribute. This results in the model having two outputs: the generated features x' that are used for verification and the predicted attribute values $\hat{a}$. Different loss values are derived from the separate outputs, corresponding to a particular task. The loss L_F derives from the extracted features x' when trained using Metric Learning for the verification/re-identification task. Whereas L_A is the penalisation loss, derived from the predicted attribute values $\hat{a}$ and the ground-truth values a. The precise definitions of L_F and L_A can be found in Sect. 3.3.

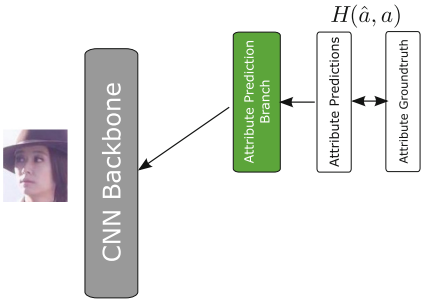
Given L_F and L_A likely have different numerical properties such as scale, stability, etc., it's necessary to weight L_F and L_A when summing them together to produce the overall loss L. To do this, like in [5], we use the regularisation hyper-parameter λ as shown in Eq. 1. Naturally, when $\lambda = 0$ the training is no longer multi-task as $L = L_F$. We use this as a baseline for both verification/re-identification and attribute discrimination performance.

$$L = (1 - \lambda)L_F + \lambda L_A \tag{1}$$

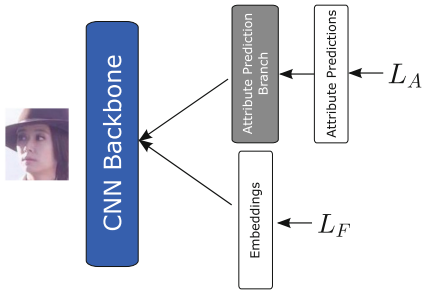
3.2 Two-Stage Adversarial Training

As stated above (Sect. 3.1) we use a multi-task learning model with a penalised loss L_A on the attribute prediction branch. However, this by itself is insufficient for the model to learn an effective embedding space while encoding no information about the protected attribute: preliminary experiments trained in a straightforward multi-task scenario failed with regards to obfuscation. The reason is that during back-propagation, the weights of the APB P will simply be updated to produce near-random output regardless of input due to the effect of L_A. Therefore the gradients penalising the encoding of the protected attribute, back-propagated into the CNN Backbone F, will be relatively insignificant and attribute obfuscation will not take place.

To ensure the penalised gradients from L_A are adequately back-propagated throughout the entire model, we employ a two-stage training step as shown in Fig. 1, with the full algorithm in Algorithm 1. This is somewhat analogous to the training of the *discriminator* and *generator* in GAN [7] based architectures.



(a) Stage 1 Back-prop: APB is trained using predicted and ground-truth attribute values with $H(\hat{a}, a)$. The CNN Backbone is not optimized.



(b) Stage 2 Back-prop: The CNN Backbone is trained with L_A and L_F (Sec. 3.3), weighted according to Eq. 1. The APB is not optimised.

Fig. 1. Two-stage training step

Stage 1. The Attribute Prediction Branch (APB) P of the Multitask model is trained using the cross-entropy between predicted and ground-truth attribute values ($H(\hat{a}, a)$, see Eq. 2) to correctly discriminate the attribute. During back-prop only the APB's parameters are updated while keeping the CNN Backbone's weights frozen. See Fig. 1a.

$$H(\hat{a}, a) = -\sum_{x \in \mathcal{X}} \hat{a}(x) \log a(x) \tag{2}$$

Stage 2. The CNN Backbone is trained with respect to L_F and L_A in a multi-task scenario. Importantly, the gradients are derived with respect to the parameters in the CNN backbone F only: the APB P is not updated during this stage. See Fig. 1b.

Algorithm 1: Two-Stage Adversarial Training

Data: F: CNN Backbone Model, P: Attribute Prediction Branch, λ,
x: images, y: identity labels, a: attribute labels,
θ_F: F's weights, θ_A: A's weights
Result: Updated θ_F, θ_A

for *epochs* **do**
  for $x_b, y_b, a_b \in (x, y, a)$ **do**
    `// Stage 1`
    $\hat{a_b} \longleftarrow P(F(x^b))$
    $\min H(\hat{a_b}, a_b)$, updating θ_A only;
    `// Stage 2`
    $x'_b \longleftarrow F(x_b)$
    $\hat{a_b} \longleftarrow P(x'_b)$
    $L \longleftarrow (1-\lambda)L_F(x'_b, y_b) + \lambda L_A(\hat{a_b}, a_b)$
    $\min L$, updating θ_F only
  end for
end for

3.3 Losses

Semi-Hard Online Mined Triplet Loss. For L_F we use the Semi-Hard Online Mined Triplet Loss described in [14]. Equation 3 shows the formulation of the triplet loss where a, p, and n are the *anchor*, *positive* and *negative* exemplars of the triplet respectively. $d(x_i, y_i) = ||\mathbf{x}_i - \mathbf{y}_i||_2$ and α is the margin (we set it to 1.0).

$$L_T(a, p, n) = \max\{d(a_i, p_i) - d(a_i, n_i) + \alpha, 0\} \tag{3}$$

If all possible valid triplets are generated, there will be many that are trivial: where $L_T(a, p, n) = 0$. Therefore the selection of valid triplets is a critical component for efficient training. At run-time, *hard* triplets where $L_T(a, p, n) > 0$ are "mined" from the minibatch. In this particular variation, priority is given to mining hard triplets which satisfy the condition $d(a_i, p_i) < d(a_i, n_i)$. These are known as the *semi-hard* triplets, as the *negative* sits within the margin α, as opposed to being closer to the *anchor* that the *positive*. It's reported in [14] that simply selecting the hardest triplets can lead to bad local minima early in training, whereas selecting these less hard triplets helps avoid that issue.

KL-Divergence with Discrete Uniform Target Distribution. The purpose of the loss L_A is to obfuscate the attribute so that no information about the attribute may be extracted from the embeddings produced by the *feature*

extractor part of the multi-task model. More concretely, when we train the CNN Backbone F we desire the output of the APB P to be random (where each predicted value is equally likely regardless of input), indicating that a discriminatory representation cannot be found.

For this reason the loss we use is a variation of the KL-Divergence shown in Eq. 4 which is a measure of relative entropy between a probability distribution Q to another probability distribution P (with both P and Q defined on the same probability space $\mathcal{X}$).

We set P to be the discrete uniform distribution $\mathcal{U}\{0, C\}$ where C is the number of discrete classes the attribute may take, with the probability of each value simply being C^{-1}. Q is set to the predicted attribute probabilities $\hat{a}$ (see Eq. 5). Given that all values in $\mathcal{U}\{0, C\}$ have an equal probability, we can further simplify this using the scalar value C^{-1} as shown in Eq. 6. Optimising this loss means the attribute predictions $\hat{a}$ move closer to a random distribution.

Using this penalising loss, in conjunction with an accurate APB P in the multi-task model (Sect. 3.1) and the two-stage training step (Sect. 3.2), we try to force the CNN Backbone F to produce embeddings which an otherwise effective attribute discriminating model cannot discriminate. This indicates no information regarding the protected attribute is encoded the in feature vectors.

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \tag{4}$$

$$L_A = D_{KL}(\mathcal{U}\{0, C\}||\hat{a}) \tag{5}$$

$$= \sum_{\hat{a}^i \in \hat{a}} C^{-1}(\log C^{-1} - \log \hat{a}^i) \tag{6}$$

4 Experimental Design

4.1 Models

In these experiments the CNN backbone F_λ, as in [14], is based on Resnet50 [8]. The Attribute Prediction Branch P_λ of the Multi-Task model, as well as the Attribute Extraction Models (AEMs) $A_\lambda^{\mathcal{D}}$, used to simulate a malicious attribute extraction and in testing attribute obfuscation performance (Sect. 4.3), are Multilayer Perceptrons with 3 Fully Connected Layers of lengths 128, 32, and C, where C is the number of classes the protected attribute may take on. Since gender is used in this work as a binary attribute, $C = 2$. The output feature vectors from F_λ are of length 512.

4.2 Datasets

We train and evaluate on VGGFace2 [4] and CelebA [10]. VGGFace2 comprises 3.31 million facial images of 9131 subjects, split into two designated partitions: *train* (8631 identities, 3138924 images) and *test* (500 identities, 169177 images).

VGGFace2 only comes with annotations for pose and age. Additional attributes were obtained from [16]. VGGFace2's gender split is 59% Male to 41% Female. Whenever VGGFace2's *train* set is used for training a model, 5% of identities are randomly selected and held out as a validation set. CelebA consists of 202599 facial images of 10177 identities, split into three designated partitions: *train* (8192 identities, 162770 images), *validation* (985 identities, 19967 images), and *test* (1000 identities, 19962 images). The images in CelebA are also annotated with 45 attributes. The gender split in CelebA is 42% Male, 58% Female. Both datasets were pre-processed using MTCNN [19] for facial alignment.

4.3 Testing Methodology

Verification Performance. We assess this by running a verification scenario using the embeddings produced by the *feature extractor* portion F_λ of the multi-task model, where λ is the value of λ during the original multi-task training (see Sect. 3.1). Evaluation takes place on a dataset's designated *test* partition. Positive pairs of images (where images belong to the same identity) are selected from the dataset, along with an equal number of non-matching images pairs (where the pair of images belong to different identities). As many unique, positive pair combinations are selected as possible, up to a limit of 1000000. This results in a balanced verification dataset of up to 2000000 matching and non-matching pairs. These image pairs are then processed by the *feature extractor* to obtain embedding pairs.

10-fold Cross-validation is used to evaluate verification performance on the embedding pairs from the testing split as described above, with 90% of pairs per fold used to calculate a optimum threshold distance using Receiver Operating Characteristic (ROC) Curves. The final 10% of embedding pairs per fold are thresholded accordingly to produce matching/non-matching label predictions for those pairs. The predicted and ground-truth labels are then used to calculate relevant metrics for that fold. The final results are the metrics averaged over all 10 folds. In addition to ROC-AUC (ROC-Area Under Curve) and accuracy, we also report False Acceptance Rate (FAR) and False Rejection Rate (FRR) as given in Eq. 7, where TP, FN, TN, FP are true-positives, false-negatives, true-negatives, and false-positives respectively.

$$\begin{aligned} FAR &= \frac{FR}{FP + TN} \\ FRR &= \frac{FN}{FN + TP} \end{aligned} \tag{7}$$

Re-identification Performance. Using a particular dataset's designated test partition, we construct a *gallery* by randomly sampling a single image per identity in the dataset. All other images from all identities are added the *probe* set. Images are passed through the *feature extractor* F_λ to obtain embeddings x'_λ, and distances between each probe embedding and all gallery embeddings are calculated and ranked. The results are averaged over all probe samples and cumulative accuracy per rank is reported.

Attribute Obfuscation Performance. For a protected attribute to be obfuscated, it means there is no information in the output embedding vectors that can be used to discriminate the attribute. To determine obfuscation performance, we run multiple attack scenarios with varying prior-information conditions. The metric we are primarily concerned with is "Balanced Accuracy" [3], where the accuracy is weighted per sample according to the inverse support of the attribute's label. Therefore, for any binary attribute a balanced accuracy value of 0.5 equates with randomised output, even on unbalanced data.

Full Technical Knowledge: The attacker has full knowledge of the models (but not the trained models themselves) and datasets used, including the corresponding attribute label for each stolen embedding. In this scenario, the attacker trains a separate model called an *Attribute Extraction Model (AEM)* using the stolen embeddings x'_λ originating from a feature model F_λ. This kind of attack would produce a feature-abusing model that can be used when further embeddings are exfiltrated without the corresponding attribute labels. One real-world situation that could enable such an attack would be the case of unsecured cloud-based storage that supports a facial verification system, containing both input (images) and output (embedding) files.

A successful attack is when the trained AEM has sufficient discriminating power to accurately predict the attribute label from the embeddings. Therefore, a successful defence would be where an AEM fails to learn such a mapping, indicating that there's insufficient latent information regarding the attribute encoded in the embeddings. Training involves predicting the attribute $\hat{a}$, and updating the AEM's weights to minimise the cross-entropy loss $H(\hat{a}, a)$ (Eq. 2).

We evaluate the AEM's performance on same and cross-dataset scenarios to help rule out results caused by overfitting to the training dataset. Concretely, each feature extractor F_λ is trained on VGGFace2's *train* set (162770 samples, 8631 identities) as part of a multi-task training scenario (see Sect. 3.1). For each F_λ, we train an AEM using embeddings generated from the same VGGFace2 *train* set A^V_λ. Training lasts for 10 epochs. For each A^V_λ (where V indicates the AEM was trained on embeddings from VGGFace2), we evaluate attribute discrimination performance using both VGGFace2's *test* set (169177 samples, 500 identities), and CelebA's *test* set (19,967 samples, 1000 identities). This results in 2 sets of results per F_λ

Partial Technical Knowledge: In this scenario, the attacker has access to the trained model F as well as stolen embeddings x' generated by F. They do not however know the attribute labels for each embedding, nor have access to the original dataset. This makes the attack in the Full Knowledge scenario impossible as supervised learning utilising the stolen embeddings x' and attribute labels a cannot be done. To get around this, the attacker uses their own annotated dataset $\mathcal{D}$ and the model F to generate new embeddings $x'^{\mathcal{D}}$. Now, as in the Full Knowledge scenario, they can train an AEM $A^{\mathcal{D}}$ using $x'^{\mathcal{D}}$ and $a^{\mathcal{D}}$ in a supervised manner, which can then be used to discriminate the gender of the original

stolen embeddings x'. With the rise of publicly available, pre-trained models this scenario may become commonplace as facial embedding models get reused across many systems. This scenario also applies to cloud computing providers which provide such models as-a-service.

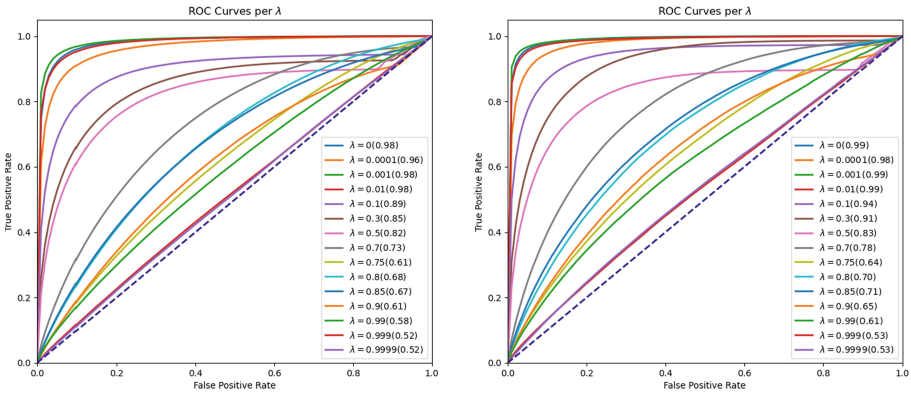
To reiterate, each model F_λ is trained on VGGFace2's *train* set. For this scenario, we use CelebA (denoted with C) as our "external" dataset. Using F_λ we generate embeddings x'^C_λ from CelebA's *train* partition, which along with a^C we use to train an AEM A^C_λ. As above, we then evaluate A^C_λ's discriminative performance on both VGGFace2 and CelebA's respective *test* partitions, giving two sets of results per F_λ.

Zero Technical Knowledge: In this scenario, the attacker has no prior technical knowledge: the only asset they have are the stolen embeddings x'. In the specific case of a binary gender attribute, given the intrinsic role gender plays in facial recognition [2] an attacker could reasonably assume that gender-based clusters exists in the embedding space. Therefore, they perform unsupervised clustering on the embeddings to assign each embedding to 1 of 2 clusters. Afterwards, using publicly available information (such as the rough demographic makeup of employees) or a reasonable guess (certain industries such as construction or the military are gender imbalanced in general), they can assign an attribute label to each cluster.

To evaluate this scenario, we generate embeddings x'_λ from each trained CNN Backbone F_λ. We reduce x'_λ to 2 Dimensions x''_λ using t-SNE [11] (with perplexity = 50) and finally cluster x''_λ with a Gaussian Mixture Model (GMM). As the goal of this attack is to simply cluster and assign attribute labels to the stolen embeddings x', and there is no danger of overfitting to the attribute labels during training as they are never used in this scenario, we train the GMM and evaluate on the same data: embeddings generated by F_λ from VGGFace2 and CelebA's test sets, producing two sets of results for each F_λ model. Note that due to the computational complexity of t-SNE, the number of images taken from each dataset are limited to a randomly sampled 10000.

4.4 Hyper-parameters

As the results in this work will primarily be judged relative to our own baseline ($\lambda = 0$, see Eq. 1), for computational efficiency we are limiting any hyper-parameter searching to λ itself. We initially test 12 values of λ: 0, 0.0001, 0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99, 0.999 and 0.9999. This choice reflects a good range of values with more precision nearer the equilibrium point while still checking many orders of magnitude. Additional values of λ were tested (0.75, 0.8, 0.85) after the initial 12 values, with the aim to find a balance between verification and attribute obfuscation performance. The optimizers used are all Adam [9] with learning rate 0.01, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The Multi-Task Models (consisting of F_λ and P_λ) and the AEMs $A^{\mathcal{D}}_\lambda$ are trained for 30 and 10 epochs respectively.



(a) VGGFace2 (b) CelebA

Fig. 2. Verification scenario results - receiver operating characteristics (ROC) curves per λ value. ROC-AUC in parentheses.

5 Results

Verification Performance: we can clearly see in Fig. 2 and Table 1 that as λ increases, it becomes increasingly difficult to use the embeddings produced by the corresponding CNN Backbone F_λ for verification. $\lambda < 0.1$ has a near-negligible

Table 1. Verification Scenario Results. Embeddings generated from dataset $\mathcal{D}$'s *test* partition by appropriate F_λ.

λ	VGGFace2				CelebA			
	AUC	Accuracy	FAR	FRR	AUC	Accuracy	FAR	FRR
0	0.982	0.936	0.064	0.064	0.990	0.955	0.045	0.045
0.0001	0.964	0.906	0.094	0.094	0.978	0.927	0.073	0.073
0.001	0.986	0.945	0.055	0.055	0.992	0.961	0.039	0.039
0.01	0.981	0.932	0.068	0.068	0.988	0.951	0.049	0.049
0.1	0.894	0.846	0.154	0.154	0.937	0.884	0.116	0.116
0.3	0.847	0.797	0.203	0.203	0.912	0.838	0.162	0.162
0.5	0.818	0.776	0.224	0.224	0.830	0.792	0.208	0.208
0.7	0.730	0.675	0.325	0.325	0.781	0.715	0.285	0.285
0.75	0.612	0.579	0.421	0.421	0.645	0.603	0.397	0.397
0.8	0.675	0.628	0.372	0.372	0.702	0.649	0.351	0.351
0.85	0.668	0.628	0.372	0.372	0.714	0.658	0.342	0.342
0.9	0.612	0.589	0.411	0.411	0.651	0.616	0.384	0.384
0.99	0.584	0.558	0.442	0.442	0.611	0.582	0.418	0.418
0.999	0.520	0.514	0.486	0.486	0.531	0.523	0.477	0.477
0.9999	0.516	0.510	0.490	0.490	0.534	0.526	0.474	0.474

effect whereas $\lambda > 0.9$ approaches near-random outputs. This matches our intuition given the adversarial nature of the multi-task learning (Sect. 3.1), and indicates that selecting an appropriate λ depends on the characteristics of the task at hand. For example, if accurate verification is the top priority then λ must be constrained appropriately, whereas if avoiding feature-abuse is paramount then a sacrifice in verification performance may be acceptable. Performance when evaluating on CelebA is slightly better (1–5% higher accuracy) across all values of λ. Considering CelebA is "wider" with many identities and fewer images each (19867 samples for 1000 identities in the test set) while VGGFace2 is"deeper" with fewer identities and more images per identity (169177 samples for 500 identities), this makes sense as we expect the intra-identity variation in VGGFace2 to be higher than in CelebA, making verification more difficult.

Table 2. 10-Identity Re-Identification Scenario. Gallery and Probe constructed from dataset $\mathcal{D}$'s *test* partition, with embeddings generated by appropriate F_λ.

λ	VGGFace2			CelebA		
	Rank 1	Rank 3	Rank 5	Rank 1	Rank 3	Rank 5
0	0.937	0.997	0.999	0.846	0.979	1.000
0.0001	0.870	0.993	0.997	0.709	0.850	0.949
0.001	0.951	0.997	1.000	0.816	0.970	0.987
0.01	0.929	0.997	0.999	0.774	0.868	0.949
0.1	0.751	0.962	0.984	0.547	0.825	0.953
0.3	0.676	0.938	0.962	0.534	0.765	0.808
0.5	0.614	0.868	0.922	0.551	0.752	0.897
0.7	0.279	0.600	0.799	0.346	0.581	0.722
0.75	0.196	0.442	0.606	0.205	0.457	0.658
0.8	0.192	0.515	0.765	0.278	0.607	0.692
0.85	0.233	0.569	0.746	0.278	0.526	0.714
0.9	0.156	0.439	0.648	0.209	0.491	0.701
0.99	0.126	0.349	0.588	0.269	0.509	0.654
0.999	0.119	0.342	0.534	0.150	0.342	0.551
0.9999	0.127	0.323	0.514	0.167	0.376	0.577

Re-identification Performance: The 10-identity scenario in Table 2 shows that performance degrades quickly after λ exceeds the equilibrium point of 0.5 and becomes almost random beyond 0.99, indicating that re-identification is more sensitive to our method than verification. That this is the case implies the least privilege learning is operating strongly on the local scale, as "neighbouring" identities in the embedding space begin to overlap significantly with sufficient values of λ.

Table 3. Attribute Extraction Attack Results. Full and Zero Knowledge Scenarios. Values are "Balanced Accuracy". 0.5 corresponds to perfect *obfuscation*. 2nd row indicates the evaluation dataset.

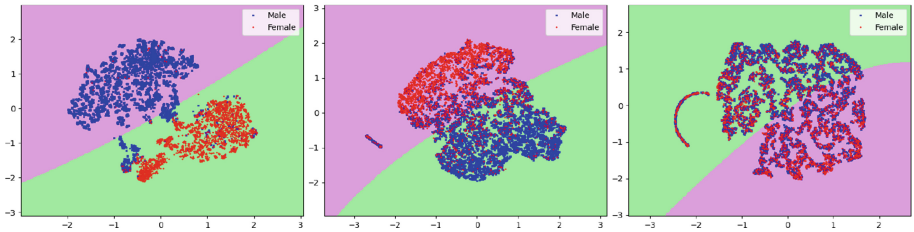
λ	Full knowledge		Partial knowledge		Zero knowledge	
	VGGFace2	CelebA	VGGFace2	CelebA	VGGFace2	CelebA
0	0.781	0.986	0.779	0.985	0.932	0.878
0.0001	0.786	0.985	0.786	0.984	0.978	0.977
0.001	0.779	0.985	0.783	0.985	0.958	0.969
0.01	0.782	0.983	0.785	0.983	0.920	0.934
0.1	0.785	0.973	0.789	0.972	0.932	0.891
0.3	0.756	0.948	0.725	0.881	0.903	0.729
0.5	0.742	0.931	0.721	0.907	0.889	0.912
0.7	0.695	0.866	0.662	0.809	0.815	0.825
0.75	0.503	0.513	0.445	0.501	0.511	0.554
0.8	0.609	0.701	0.592	0.669	0.539	0.525
0.85	0.613	0.753	0.607	0.738	0.579	0.545
0.9	0.498	0.500	0.444	0.500	0.574	0.546
0.99	0.497	0.500	0.444	0.500	0.516	0.528
0.999	0.497	0.500	0.448	0.504	0.529	0.506
0.9999	0.497	0.500	0.444	0.500	0.522	0.524

Attribute Obfuscation (Full Knowledge): The attribute discrimination performance of the AEMs A_λ^V have no significant deterioration until $\lambda >= 0.3$, after which performance declines until the attributes are sufficiently obfuscated at $\lambda = 0.75$ and $\lambda >= 0.9$. See Table 3. This being the case when evaluating on both VGGFace2 and CelebA illustrates the cross-dataset viability of our method.

Attribute Obfuscation (Partial Knowledge): The performance of the AEMs A_λ^C in this scenario aligns closely with the performance of the Full Knowledge AEMs A_λ^V, with obfuscation occurring at the same values: $\lambda = 0.75$ and $\lambda >= 0.9$. That the two scenarios give similar results indicate that λ is the dominant factor, with the specific embeddings $x'_\mathcal{D}$ the AEMs are trained with playing a lesser role in discrimination performance.

Attribute Obfuscation (Zero Knowledge): The results in Table 3 show that extracting a binary gender attribute from embeddings intended for facial verification can be quite straightforward, even under unsupervised approaches with little knowledge held by the attacker. Thus, segmenting an embedding space into two clusters can result in a balanced accuracy of over 95% when λ is sufficiently small. The discriminative performance drops significantly when $\lambda > 0.5$, with the attribute being obfuscated at $\lambda >= 0.75$ so mitigating against this attack is slightly easier than the Full and Partial Knowledge scenarios, but is in no way trivial. Figure 3 visualises the results. We can see that at $\lambda = 0$ the embeddings

are distinctly clustered by gender. At $\lambda = 0.7$, the clusters have begun to merge and the gender attribute has become more diffuse, before the total obfuscation shown at $\lambda = 0.9$.



(a) $\lambda = 0$ (b) $\lambda = 0.7$ (c) $\lambda = 0.9$

Fig. 3. Visualisation of Zero Technical Knowledge attack results. Embeddings generated from VGGFace2's test set. Background colours indicate GMM Decision boundary for each cluster.

6 Conclusion

In this paper we propose a novel training paradigm to obfuscate protected attributes in verification and re-identification systems, enabling least privilege learning in the interest of stopping feature abuse. It achieves this by applying the KL-Divergence with Discrete Uniform Target Distribution loss to the protected attribute, in conjunction with a Two-Stage Adversarial Training procedure in a Multi-Task Learning scenario. In terms of obfuscating protected attributes, we succeed at reducing the amount of extractable latent information regarding the attribute in the resulting feature vectors to near zero, given sufficiently large values of λ. While the main learning task performance may suffer, an effective balance between verification/re-identification and attribute obfuscation is possible in the range of $\lambda \in [0.75, 0.9]$.

We have also proposed a comprehensive evaluation framework combining 2 different datasets and 3 application settings: verification, re-identification, and attribute prediction across multiple scenarios with various levels of attacker knowledge, that allows us to clearly measure the (often competing) performance requirements of a least privilege learning model. This could be used as a comprehensive testbench for future works.

As future work we will focus on maintaining the obfuscation performance of the proposed method, while minimising the sacrifice in identity discriminating power. Further, the application of least privilege learning would be most powerful if successfully applied to multiple attributes simultaneously.

References

1. Alvi, M., Zisserman, A., Nellåker, C.: Turning a blind eye: explicit removal of biases and variation from deep neural network embeddings. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11129, pp. 556–572. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11009-3_34
2. Baudouin, J.Y., Tiberghien, G.: Gender is a dimension of face recognition. J. Exp. Psychol. Learn. Mem. Cogn. **28**(2), 362–365 (2002)
3. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 2010 20th International Conference on Pattern Recognition, pp. 3121–3124 (August 2010)
4. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: a dataset for recognising faces across pose and age. arXiv:1710.08092 [cs] (May 2018)
5. Ganin, Y.: Domain-adversarial training of neural networks. In: Csurka, G. (ed.) Domain Adaptation in Computer Vision Applications. ACVPR, pp. 189–209. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58347-1_10
6. Georgopoulos, M., Oldfield, J., Nicolaou, M.A., Panagakis, Y., Pantic, M.: Mitigating demographic bias in facial datasets with style-based multi-attribute transfer. Int. J. Comput. Vis. **129**(7), 2288–2307 (2021). https://doi.org/10.1007/s11263-021-01448-w
7. Goodfellow, I., et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680 (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (June 2016)
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 [cs] (January 2017)
10. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. arXiv:1411.7766 [cs] (September 2015)
11. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008)
12. Quadrianto, N., Sharmanska, V., Thomas, O.: Discovering fair representations in the data domain. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, pp. 8219–8228 (June 2019)
13. Raff, E., Sylvester, J.: Gradient reversal against discrimination. arXiv:1807.00392 [cs, stat] (July 2018)
14. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (June 2015)
15. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv:1312.6199 [cs] (February 2014)
16. Terhörst, P., Fährmann, D., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: MAAD-Face: a massively annotated attribute dataset for face images. arXiv:2012.01030 [cs] (December 2020)
17. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4068–4076 (December 2015)

18. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New York, NY, USA, pp. 335–340 (December 2018)
19. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Sig. Process. Lett. **23**(10), 1499–1503 (2016)