Interpretable Privacy for Deep Learning Inference

Fatemehsadat Mireshghallah [1] Mohammadkazem Taram [1]
Ali Jalali [2] Ahmed Taha Elthakeb [1] Dean Tullsen [1] Hadi Esmaeilzadeh [1]

Abstract

In order to receive machine learning services from a cloud-based service provider, consumers usually send their entire raw data (e.g. an entire image). However, this models reveals much more information to the service provider than what is actually necessary for the execution of the service. This work shows that, in many cases, only a small portion of the input is required for the service provider to offer an accurate prediction. Discovering this subset is one of the main objectives of this paper. We formulate this problem as a gradient-based perturbation maximization method that discovers this subset in the input feature space with respect to the decision making of the prediction model used by the provider. After identifying the essential subset, our framework, Cloak, suppresses the rest of the features in the consumer's input and only sends the essential ones to the cloud. As such, the service provider can use those features to return an accurate prediction and also to improve its service, while at the same time the privacy of the consumer is better protected. We also demonstrate in our experiments that by removing the extra features, the post-hoc fairness of the classifier is improved as well.

1 Introduction

The computational complexity of Machine Learning (ML) models has pushed their execution to the cloud. The edge devices on the user side capture and send their data to the cloud for *prediction services*. The insight in this paper is that a large fraction of the data is not relevant to the prediction service and can be segregated prior to sending the data out, thus enabling access to the services with much greater privacy. As such, we propose Cloak, an orthogonal approach to the existing techniques that mostly rely on cryptographic solutions and impose prohibitive delays and computational cost. Table 1 summarizes most state-of-the-art encryption-based methods and their runtime compared to unencrypted execution on GPUs. As shown, these techniques impose between 318× to 14,000× slowdown. An image classification inference is performed in multiple seconds, an order of magnitude away from the service-level agreement between users and cloud providers, which is between 10 to 100 milliseconds according to MLPerf industry measures (Reddi et al., 2020; MLPerf Organization, 2020). Such slowdowns will lead to unacceptable interaction with services that require near real-time response (e.g., home automation cameras). Cloak provides a middle ground, where there is a provable degree of privacy while the prediction latency is essentially unaffected. To that end, Cloak only sends out the features that the provider essentially requires to carry out the requested service. Existing privacy techniques are applicable to scenarios that can tolerate longer delays, but are not currently suitable for consumer applications, which rely on interactive prediction services. However, having no privacy protection is also not desirable.

Table 1: Slowdown of cryptographic techniques vs. conventional GPU execution on Titan Xp and Cloak.

Cryptographic Technique	Release Year	Dataset	Prediction Time (sec)			Slowdown
			Encry.	Conv.	Cloak	
FALCON (Wagh et al., 2020)	2020	ImageNet	12.96	0.0145	0.0148	906×
DELPHI (Mishra et al., 2020)	2020	CIFAR-100	3.5	0.0112	0.0113	318×
CrypTen (Facebook, 2019)	2019	ImageNet	8.30	0.0121	0.0123	691×
GAZELLE (Juvekar et al., 2018)	2018	CIFAR-100	82.00	0.0112	0.0113	7,454×
MiniONN (Liu et al., 2017)	2017	MNIST	9.32	0.0007	0.0007	14,121×

This paper presents Cloak, a framework that segregates the features of the data based on their relevance to the target prediction task. To solve this problem, we reformulate the objective as a *gradient-based* optimization problem, that generates a *segregated representation of the input*. The intuition is that if a feature can consistently tolerate addition of noise without degrading the utility, that feature is not conducive to the classification task and can be suppressed. By removing such features, Cloak guarantees that no information about them can be learned or inferred from the segregated representation that the consumer sends. Figure 1 shows examples of conducive features for multiple tasks discovered by Cloak and the corresponding segregated representation for an example image. Our differentiable formulation of finding the scales minimizes the upper bound of the Mutual Information (MI) between the irrelevant features and the segregated representation (maximizing privacy) while maximizing the lower bound of MI between the relevant features and the generated representation (preserving utility).

Experimental evaluation with real-world datasets of UTKFace (Zhang & Qi, 2017), CIFAR-100 (Krizhevsky et al.), and MNIST (LeCun & Cortes) shows that Cloak can reduce the mutual information between input images and the

[1] University of California San Diego [2] Amazon. Correspondence to: Fatemehsadat Mireshghallah <fmireshg@eng.ucsd.edu>.

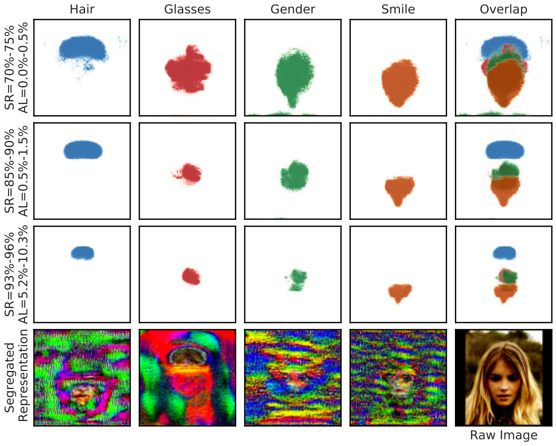


Figure 1: Cloak's discovered features for target DNN classifiers (VGG-16) for black-hair color, eyeglasses, gender, and smile detection. The colored features are conducive to the task. The 3 sets of features depicted for each task correspond to different suppression ratios (SR). AL denotes the range of accuracy loss imposed by the suppression.

publicized representation by 85.01% with accuracy loss of only 1.42%. In addition, we evaluate the protection offered by Cloak against adversaries that try to infer data properties from segregated representations on CelebA dataset (Liu et al., 2015). We show that segregated representations generated for "smile detection" as the target task effectively prevent adversaries from inferring information about hair color and/or eyeglasses. We show that Cloak can provide these interpretations and protections even in a black-box setting where we do not have access to the service provider's model parameters or architecture. We further show that Cloak can improve the classifier's fairness.

2 Cloak's Optimization Problem

This section formally describes the optimization problem and presents a computationally tractable method towards solving it. Let $\mathbf{x} \in \mathbf{R}^n$ be an input, and $\mathbf{c} \subseteq \mathbf{x}$ and $\mathbf{u} \subseteq \mathbf{x}$ be two disjoint sets of conducive and non-conducive features with respect to our target classifier (f_θ). We construct a noisy representation $\mathbf{x_c} = \mathbf{x} + \mathbf{r}$ where $\mathbf{r} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma}$ is a diagonal covariance matrix, as we set the elements of the noise to be independent. This noisy representation helps find the conducive features and is used to create a final suppressed representation $\mathbf{x_s}$ that is sent to the service provider. The goal is to construct $\mathbf{x_c}$ such that the mutual information between $\mathbf{x_c}$ and $\mathbf{u}$ is minimized (for privacy), while the mutual information between $\mathbf{x_c}$ and $\mathbf{c}$ is maximized (for utility). The is written as the following soft-constrained optimization problem:

$$\min_{\mathbf{x_c}} \quad I(\mathbf{x_c}; \mathbf{u}) - \lambda I(\mathbf{x_c}; \mathbf{c}) \qquad (1)$$

To solve this problem, we bound the terms of our optimization problem of Equation 1, and then take an iterative approach (Blundell et al., 2015). To this end, we find an upper bound for $I(\mathbf{x_c}; \mathbf{u})$ and a lower bound for $I(\mathbf{x_c}; \mathbf{c})$.

2.1 Upper bound on $I(\mathbf{x_c};\mathbf{u})$

Since $\mathbf{u}$ is a subset of $\mathbf{x}$, the following holds:

$$I(\mathbf{x_c}; \mathbf{u}) \leq I(\mathbf{x_c}; \mathbf{x}) = \mathcal{H}(\mathbf{x_c}) - \frac{1}{2}\log((2\pi e)^n |\boldsymbol{\Sigma}|) \qquad (2)$$

Where $\mathcal{H}(\mathbf{x_c}|\mathbf{x})$ is the entropy of the added Gaussian noise. Here $|\boldsymbol{\Sigma}|$ denotes the determinant of the covariance matrix. Then by applying Theorem 8.6.5 from (Cover & Thomas, 2012) which gives an upper bound for the entropy, to $\mathbf{x_c}$, we can write:

$$I(\mathbf{x_c}; \mathbf{u}) \leq \tfrac{1}{2}\log((2\pi e)^n \tfrac{|Cov(\mathbf{x_c})|}{|\boldsymbol{\Sigma}|}) \qquad (3)$$

Since $\mathbf{x}$ and $\mathbf{r}$ are independent variables and $\mathbf{x_c} = \mathbf{x} + \mathbf{r}$, we have $|Cov(\mathbf{x_c})| = |Cov(\mathbf{x}) + \boldsymbol{\Sigma}|$. In addition, since covariance matrices are positive semi-definite, we can get the eigen decomposition of $Cov(\mathbf{x})$ as $Q\Lambda Q^T$ where the diagonal matrix Λ has the eigenvalues. Since $\boldsymbol{\Sigma}$ is already a diagonal matrix, $|Cov(\mathbf{x}) + \boldsymbol{\Sigma}| = |Q(\Lambda + \boldsymbol{\sigma}^2)Q^T| = \prod_{i=1}^n (\lambda_i + \sigma_i^2)$. By substituting this in Equation 3, and simplifying we get the upper bound for $I(\mathbf{x_c}; \mathbf{u})$ as the following:

$$I(\mathbf{x_c}; \mathbf{u}) \leq \tfrac{1}{2}\log((2\pi e)^n \textstyle\prod_{i=1}^n (1 + \tfrac{\lambda_i}{\sigma_i^2})) \qquad (4)$$

2.2 Lower bound on $I(\mathbf{x_c};\mathbf{c})$

Theorem 2.1. *The lower bound on $I(\mathbf{x_c};\mathbf{c})$ is:*

$$\mathcal{H}(\mathbf{c}) + \max_q \mathbb{E}_{\mathbf{x_c},\mathbf{c}}[\log q(\mathbf{c}|\mathbf{x_c})] \qquad (5)$$

Where q denotes all members of a possible family of distributions for this conditional probability.

Proof. The lemma and accompanying proof for this theorem are redacted to save space. □

2.3 Loss Function

Now that we have the upper and lower bounds, we can reduce our problem to the following optimization where we minimize the upper bound (Equation 4) and maximize the lower bound (Equation 5):

$$\min_{\boldsymbol{\sigma}, q} \quad \tfrac{1}{2}\log((2\pi e)^n \textstyle\prod_{i=1}^n (1 + \tfrac{\lambda_i}{\sigma_i^2})) + \lambda \sum_{\mathbf{c_i}, \mathbf{x_{c_i}}} (-\log q(\mathbf{c_i}|\mathbf{x_{c_i}})) \qquad (6)$$

We write the expected value in the same equation in the form of a summation over all possible representations and conducive features. To make this summation tractable, in our loss function we replace this part of the formulation with the empirical cross-entropy loss of the target classifier over all training examples. We also relax the optimization further by rewriting the first term. Since minimizing this term is equivalent to maximizing the standard deviation of the noise, we change the fraction into a subtraction. Our final loss function becomes:

$$\mathcal{L} = -\log \tfrac{1}{n} \textstyle\sum_{i=0}^n \sigma_i^2 + \lambda \mathbb{E}_{\mathbf{r} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^{\mathbf{2}}), \mathbf{x} \sim \mathcal{D}} \left[-\textstyle\sum_{k=1}^K y_k \log(f_\theta(\mathbf{x} + \mathbf{r}))_k \right] \qquad (7)$$

The second term is the expected cross-entropy loss, over the randomness of the noise and the data instances. The variable $\boldsymbol{\mu}$ is the mean of the noise distributions. The variable K is the number of classes for the target task, and y_k is the indicator variable that determines if a given example belongs to class k. More intuitively, the first term increases the noise of each feature and provides privacy. The second term decrease the classification error and maintains accuracy. The parameter λ is a knob which provides a trade-off between these two.

2.4 Suppressed Representation

After finding the noisy representation $\mathbf{x_c}$, we use it to generate the final suppressed representation $\mathbf{x_s}$. By applying a cutoff threshold T on $\boldsymbol{\sigma}$, we generate binary mask $\mathbf{b}$ such that $b_i = 1$ if $\sigma_i \geq T$, and $b_i = 0$ otherwise. We create representation $\mathbf{x_s} = (\mathbf{x} + \mathbf{r}) \circ \mathbf{b} + \mu_\mathbf{s}$, where $\mathbf{r} \sim \mathcal{N}(0,\boldsymbol{\sigma})$ and $\mu_\mathbf{s}$ are constant values that are set to replace non-conducive features. According to the data processing inequality (Beaudry & Renner, 2011), the upper bound on $I(\mathbf{x_c};\mathbf{u})$ holds for $\mathbf{x_s}$ as well, since $I(\mathbf{x_s};\mathbf{u}) \leq I(\mathbf{x_c};\mathbf{u})$. The same inequality also implies that the lower bound achieved for $I(\mathbf{x_c};\mathbf{c})$ does not necessarily hold for $\mathbf{x_s}$. To address this, we write another optimization problem, to find $\mu_\mathbf{s}$ such that cross entropy loss, i.e, $\min_{\mu_\mathbf{s}} \sum_{k=1}^{K} y_k \log(f_\theta(\mathbf{x_s}))_k$ is minimized. Solving this guarantees that the lower bound of Equation 5 also holds for $I(\mathbf{x_s};\mathbf{c})$.

3 Cloak Framework

Cloak comprises of two phases: first, an offline phase where we solve the optimization problems and Second, an online prediction phase where we suppress the input data. In this section we discuss details of these two phases, starting from the details of the offline phase.

3.1 Noise Re-parameterization and Constraints

To solve the optimization problem of Section 2, Cloak's approach is to cast the noise distribution parameters as trainable tensors, making it possible to solve the problem using conventional gradient-based methods. To be able to define gradients over the means and variances, we rewrite the noise sampling to be $\mathbf{r} = \boldsymbol{\sigma} \circ \mathbf{e} + \boldsymbol{\mu}$, instead of $\mathbf{r} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma^2})$, where $\mathbf{e} \sim \mathcal{N}(0,1)$. The symbol $\circ$ denotes the element-wise multiplication of elements of $\boldsymbol{\sigma}$ and $\mathbf{e}$. We also need to reparameterize $\boldsymbol{\sigma}$ to limit the range of standard deviation of each feature (σ). If it is learned through a gradient-based optimization, it can take on any value, while we know that variance can not be negative. In addition, we also do not want the σs to grow over a given maximum, M. We put this extra constraint on the distributions, to limit the σs from growing infinitely (to decrease the loss), taking the growth opportunity from the standard deviation of the other features. Finally, we define a trainable parameter $\boldsymbol{\rho}$ and write $\boldsymbol{\sigma} = \frac{1.0+\tanh(\boldsymbol{\rho})}{2} M$, where the $\tanh$ function is used to constraint the range of the σs, and the addition of 1 is to guarantee the positivity of the variance.

3.2 Cloak's Perturbation Training Workflow

Algorithm 1 shows the steps of Cloak's optimization process. This algorithm takes the training data ($\mathcal{D}$), labels (y), a pre-trained model (f_θ), and the privacy-utility knob (λ) as input, and computes the optimized tensor for noise distribution parameters. During the initialization step, the algorithm sets the trainable tensor for the means ($\boldsymbol{\mu}$) to 0, and initializes the substitute trainable tensor ($\boldsymbol{\rho}$) with a large negative number. Since the loss (Equation 7) incorporates expected value over noise samples, Cloak uses Monte Carlo sampling (Kalos & Whitlock, 1986) with sufficiently large number of noise samples to calculate te loss. Once the training is finished, the optimized mean and standard deviation tensors are collected and passsed to the next phase.

3.3 Feature Segregation and Suppression

To suppress the non-conducive features one simple way is to send the noisy representations, i.e, adding noise from the $(\boldsymbol{\mu},\boldsymbol{\sigma}^2)$ to the input to get the representations that are sent out for prediction. This method, however, suffers from two shortcomings: first, it does not directly suppress and remove the features, which could leave the possibility of data leakage. Second, because of the high standard deviations of noise, in some cases the generated representation might be out of the domain of the target classifier, which could have negative effects on the utility.

Another way of suppressing the non-conducive features is to replace them with zeros (black pixels in images for example). This scheme also, suffers from potential accuracy degradation, as the values we are using for suppression (i.e. the zeros) might not match the distribution of the data that the classifier expects. To mitigate this, we find a suppressed representation, i.e., we train the constant suppression values that need to replace the non-conducive features. Intuitively, these learned values reveal what the target classifier perceives as common among all the inputs from the training set, and what it expects to see. You can see a comparison of these three schemes in the experimental results section. Algorithm 2 shows the steps of this training process. The algorithm finds $\boldsymbol{\mu}_s$, the values by which we replace the non-conducive features. The only objective of this training process is to increase the accuracy, therefore we use cross-entropy loss as our loss function.

Algorithm 1 Noise Train.

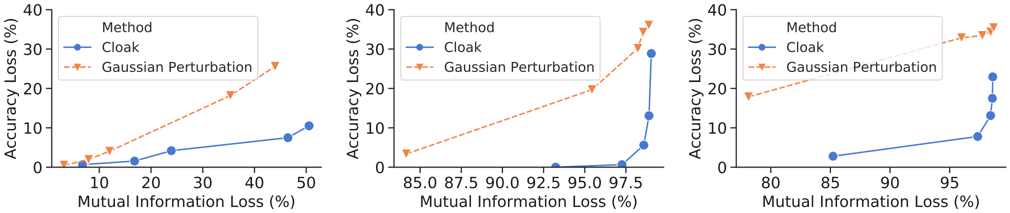
```
Input: $\mathcal{D}$, $y$, $f_\theta$, $m$, $\lambda$
Initialize $\boldsymbol{\mu} = 0$, $\boldsymbol{\rho} = -10$, $M \geq 0$
repeat
    Select training batch $\mathbf{x}$ from $\mathcal{D}$
    Sample $\mathbf{e} \sim \mathcal{N}(0,1)$
    Let $\boldsymbol{\sigma} = \frac{1.0+\tanh(\boldsymbol{\rho})}{2}(M)$
    Let $\mathbf{r} = \boldsymbol{\sigma} \circ \mathbf{e} + \boldsymbol{\mu}$
    Gradient step on $\boldsymbol{\mu}$, $\boldsymbol{\rho}$ from Eq. (7)
until Algorithm converges
Return: $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$
```

Algorithm 2 Suppr. Train.

```
Input: $\mathcal{D}$, $y$, $f_\theta$, $\boldsymbol{\sigma}$, $\boldsymbol{\mu}$, $\mathbf{b}$
Initialize $\boldsymbol{\mu_s} = \boldsymbol{\mu}$
repeat
    Select training batch $\mathbf{x}$ from $\mathcal{D}$
    Sample $\mathbf{r} \sim \mathcal{N}(0,\boldsymbol{\sigma}^2)$
    Let $\mathbf{x_s} = (\mathbf{x}+\mathbf{r}) \circ b + \boldsymbol{\mu_s}$
    Take gradient step on $\boldsymbol{\mu}_s$ from
    $\mathbb{E}_r[\mathcal{L}_{CE}(f_\theta(\mathbf{x_s}), y)]$
until Algorithm converges
Return: $\boldsymbol{\mu}_s$
```

3.4 Online Prediction

The prediction (inference) phase is when unseen test inputs that we protect are sent to the remote service provider



(a) CIFAR-100

(b) MNIST

(c) UTKFace

Figure 2: Privacy-accuracy trade-off.

for classification. First a noise tensor sampled from the optimized distribution $\mathcal{N}(0,\boldsymbol{\sigma}^2)$ is added to the input, then the binary mask b is applied to the noisy input image. Finally $\boldsymbol{\mu}_s$ is added to $\mathbf{x}$ and the resulting segregated representation is sent to the service provider. As an example, the last row of Figure 1 shows representations generated by Cloak.

4 Experimental Results

To evaluate Cloak, we use four real-world datasets on four Deep Neural Networks (DNNs). Namely, we use VGG-16 (Simonyan & Zisserman, 2014) and ResNet-18 (He et al., 2016) on CelebA (Liu et al., 2015), AlexNet (Krizhevsky et al., 2012) on CIFAR-100 (Krizhevsky et al.), a modified version of VGG-16 model on UTKFace (Zhang & Qi, 2017), and LeNet-5 (LeCun, 1998) on MNIST (LeCun & Cortes). The mutual information numbers reported in this section are estimated over the test set using the Shannon Mutual Information estimator provided by the Python ITE toolbox (Szabó, 2014).

4.1 Privacy-Accuracy Trade-Off

Figure 2 shows accuracy loss of the DNN classifiers using segregated representations vs. the loss in mutual information. This is the loss in mutual information between the original image and its noisy representation, divided by the amount of information in bits in the original image. In this experiment, we compare Cloak to adding Gaussian perturbation of mean zero and different standard deviations to all pixels of the images. For fair comparison, we choose Cloak's suppression with noisy representations. For MNIST and UTKFace, Cloak reduces the information in the input significantly (93% and 85% respectively) with little loss in accuracy (0.5% and 2.7%).

4.2 Adversary to Infer Information

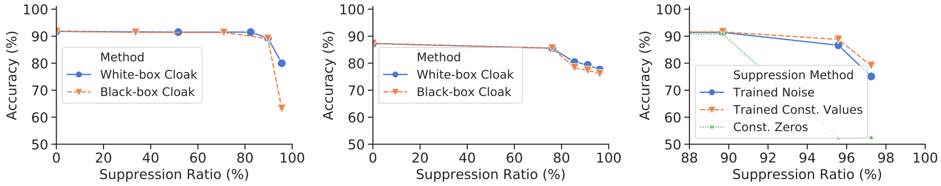
To further evaluate the effectiveness of the representations that Cloak generates, we devise an experiment in which an adversary tries to infer properties of the segregated representations using a DNN classifier. We assume two adversary models here. First, the adversary has access to a unlimited number of samples from the segretaed representations, therefore she can re-train her classifier to regain accuracy on the segregated representations. Second, a model in which the adversary's access to the segregated representation is limited and therefore she cannot retrain her classifier on the segregated representations. In this experiment, we choose smile detection as the target prediction task for which Cloak generates representations. Then, we model adversaries who try to discover two properties from the segregated representations: whether people in images wear glasses or not and whether their hair is black or not. The adversaries have pre-trained classifiers for both these tasks.

Figure 4 shows the results of this experiment. Each point in this figure is generated using a noise map with a Suppression Ration (SR) noted in the figure. Higher SR means more features are suppressed. When adversaries do not retrain their models, using segregated representations with 95.6% suppression ratio causes the adversaries to almost completely lose their ability to infer eyeglasses or hair color and reach to the random classifier accuracy (50%). This is achieved while the target smile detection task only loses 5.16% accuracy. When adversaries retrain their models, using representations with slightly higher suppression ratio (98.3%) achieves the same goal. But this time, the accuracy of the target task drops to 78.9%. With the same suppression ratio, the adversary who tries to infer hair color loses more accuracy than the adversary who tries to infer eyeglasses. This is because, as shown in Figure 1, the conducive features of smile overlap less with the conducive features of hair than with the conducive features of eyeglasses.

4.3 Black-Box Access Mode

To show the applicability of Cloak, we show that it is possible for Cloak to protect users' privacy even when we have limited access to the target model. We consider a black-box setting in which we assume Cloak does not have any knowledge of the target model architecture or its parameters and is only allowed to send requests and get back responses. In this setting, following similar methodology to the methodology described in Shokri et al. (Shokri et al., 2017)we first train a substitute model that helps us to train Cloak's representations. We assume a target service provider that has two ResNet18 (He et al., 2016) DNNs deployed, one for the task of black hair color classification, and one for smile detection. Since we assume no knowledge of the model architecture, Cloak substitutes the target classifiers with another architecture, i.e, with two VGG-16 DNNs. Cloak substitute models for the hair and smile tasks have accuracies of 84.9% and 90.9% and the target models have accuracies of 87.3% and 91.8%. After training the substitute model, we apply Cloak to them to find noise maps and suppressed representations.

Figure 3a and 3b show the results for these experiments.



(a) Black-box-smile (b) Black-box-hair color (c) Suppression Schemes

Figure 3: (a) and (b) performance of Cloak in a black-box setting and (c) the effect of different suppression schemes

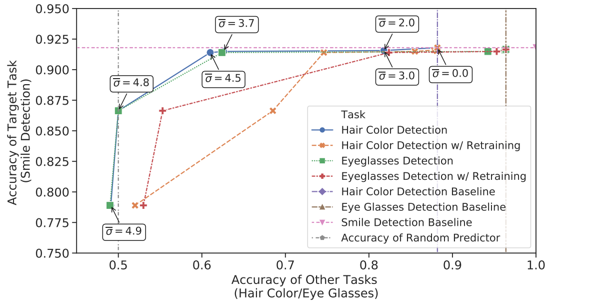


Figure 4: Protection from adversaries that infer black-hair color or eyeglasses from the segregated representations.

Cloak performs similarly effective in both white-box and black-box settings and for both hair color classification and smile detection tasks. The reason is that the DNN classifiers of the same task are known to learn similar patterns and decision boundaries (Papernot et al., 2017; Arpit et al., 2017).

4.4 Post-hoc Effects of Cloak on Fairness

Cloak, by removing extra features, not only benefits privacy but can also remove unintended biases of the classifier, resulting in a more fair classification. In many cases the features that bias the classifiers highly overlap with the non-conducive features that Cloak discovers. Therefore, applying Cloak can results in the predictions that are more fair, without the need to change the classifier. This subsection evaluates this positive side-effect of Cloak by adopting a setup similar to that of Kairouz et al. (Kairouz et al., 2019). We measure the fairness of the black-hair color classifier using the segregated representations, while considering gender to be a sensitive variable that can cause bias. We use two metrics for our experiments, the difference in Demographic Parity (Δ_{DemP}), and the difference in Equal Opportunity (Δ_{EO}). Figure 5 shows that as Cloak suppresses more non-conducive features, the fairness metrics improve significantly. We see 0.05 reduction in both metrics due to the removal of gender related non-conducive features. It is noteworthy that the biasing features in the hair color classifier are not necessarily the gender features shown in Figure 1. Those features show what a gender classifier uses to make its decision.

4.5 Different suppression schemes

Figure 3c shows the accuracy of three suppression schemes described in Section 3.3 on the smile detection task (on CelebA/ VGG-16). Among different schemes, suppression using the trained values yields better accuracy for the same suppression ratio, since it captures what the classifier expects to receives. Suppression with noise (sending noisy representations) performs slightly worse than training, and that is mainly due to the uncertainty brought by the noise.

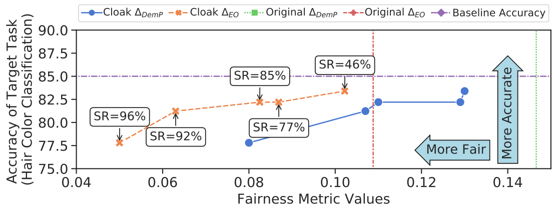


Figure 5: Shows effects of Cloak on post-hoc fairness.

5 Related Work

For *training*, the literature abounds with the studies that use noise addition as a randomization mechanism to protect privacy (Chaudhuri et al., 2009; 2013; Dwork & Roth, 2014; Abadi et al., 2016; Shokri & Shmatikov, 2015; Papernot et al., 2016; 2018). Privacy on offloaded computation can also be provided by the means of cryptographic tools such as homomorphic encryption and/or Secure Multiparty Computation (SMC) (Hesamifard et al., 2017; Juvekar et al., 2018; Mohassel & Zhang, 2017; Dowlin et al., 2016; Liu et al., 2017; Mishra et al., 2020; Wagh et al., 2020; Agrawal et al., 2019). However, these approaches suffer from a prohibitive computational costs (See Table 1), on both cloud and user side, exacerbating the complexity and compute-intensivity of neural networks especially on resource-constrained edge devices. Only a handful of studies have addressed privacy of prediction by adding noise to the data (Osia et al., 2020; Mireshghallah et al., 2020). Shredder (Mireshghallah et al., 2020) proposes to *heuristically* sample and reorder additive noise at run time based on the previously collected additive tensors that the DNN can tolerate (anti-adversarial patterns). Due to the heuristic and pattern-based nature of this prior work, it does not provide formal guarantees. In contrast, Cloak's approach is to directly learn conducive features and suppress non-conducive ones with learned constant values.

6 Conclusion

The surge in the use of machine learning is driven by growth in data and compute power. The data mostly comes from people (Thompson & Warzel, 2019) and includes an abundance of private information. We propose Cloak, a mechanism that finds features in the data that are unimportant and non-conducive for a cloud ML prediction model.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

Agrawal, N., Shahin Shamsabadi, A., Kusner, M. J., and Gascón, A. Quotient: two-party secure neural network training and prediction. In *ACM Conference on Computer and Communications Security (CCS)*, 2019.

Arpit, D., Jastrzkebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Beaudry, N. J. and Renner, R. An intuitive proof of the data processing inequality. *arXiv preprint arXiv:1107.0740*, 2011.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *arXiv preprint arXiv:0912.0071*, 2009.

Chaudhuri, K., Sarwate, A. D., and Sinha, K. A near-optimal algorithm for differentially-private principal components. *J. Mach. Learn. Res.*, 14(1):2905–2943, January 2013. ISSN 1532-4435.

Cover, T. M. and Thomas, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning (ICML)*, 2016.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9:211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL *http://dx.doi.org/10.1561/0400000042*.

Facebook. A research tool for secure machine learning in pytorch, 2019. online–accessed June 2020, url: *https://crypten.ai*.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hesamifard, E., Takabi, H., and Ghasemi, M. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*, 2017. URL *http://arxiv.org/abs/1711.05189*.

Juvekar, C., Vaikuntanathan, V., and Chandrakasan, A. GAZELLE: A low latency framework for secure neural network inference. In *USENIX Security Symposium (USENIX Security)*, 2018.

Kairouz, P., Liao, J., Huang, C., and Sankar, L. Censored and fair universal representations using generative adversarial models. *arXiv preprint arXiv:1910.00411*, 2019.

Kalos, M. H. and Whitlock, P. A. *Monte Carlo Methods. Vol. 1: Basics*. Wiley-Interscience, USA, 1986. ISBN 0471898392.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-100 (canadian institute for advanced research). URL *http://www.cs.toronto.edu/~kriz/cifar.html*. url: *http://www.cs.toronto.edu/~kriz/cifar.html*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60:84–90, 2012.

LeCun, Y. Gradient-based learning applied to document recognition. 1998.

LeCun, Y. and Cortes, C. The mnist dataset of handwritten digits. online accessed May 2019 *http://www.pymvpa.org/datadb/mnist.html*.

Liu, J., Juuti, M., Lu, Y., and Asokan, N. Oblivious neural network predictions via minionn transformations. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, 2015.

Mireshghallah, F., Taram, M., Ramrakhyani, P., Jalali, A., Tullsen, D., and Esmaeilzadeh, H. Shredder: Learning noise distributions to protect inference privacy. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020.

Mishra, P., Lehmkuhl, R., Srinivasan, A., Zheng, W., and Popa, R. A. Delphi: A cryptographic inference service for neural networks. In *USENIX Security Symposium (USENIX Security)*, 2020. URL *https://www.usenix.org/conference/usenixsecurity20/presentation/mishra*.

MLPerf Organization. MLPerf Benchmark Suite, 2020. url: *https://mlperf.org*.

Mohassel, P. and Zhang, Y. Secureml: A system for scalable privacy-preserving machine learning. In *IEEE Symposium on Security and Privacy (S&P)*, 2017.

Osia, S. A., Shamsabadi, A. S., Sajadmanesh, S., Taheri, A., Katevas, K., Rabiee, H. R., Lane, N. D., and Haddadi, H. A hybrid deep learning architecture for privacy-preserving mobile analytics. *IEEE Internet of Things Journal*, pp. 1–1, 2020. ISSN 2372-2541. doi: 10.1109/JIOT.2020.2967734.

Papernot, N., Abadi, M., Úlfar Erlingsson, Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *ACM on Asia conference on computer and communications security (AsiaCCS)*, 2017.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C.-J., Anderson, B., Breughe, M., Charlebois, M., Chou, W., et al. MLPerf inference benchmark. In *International Symposium on Computer Architecture (ISCA)*, 2020.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *ACM Conference on Computer and Communications Security (CCS)*, 2015.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (S&P)*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szabó, Z. Information theoretical estimators toolbox. *Journal of Machine Learning Research*, 15:283–287, 2014.

Thompson, S. A. and Warzel, C. The privacy project: Twelve million phones, one dataset, zero privacy, 2019. online–accessed February 2020, url: *https://www.nytimes.com/interactive/2019/12/19/opinion/location-tracking-cell-phone.html*.

Wagh, S., Tople, S., Benhamouda, F., Kushilevitz, E., Mittal, P., and Rabin, T. Falcon: Honest-majority maliciously secure framework for private deep learning. *arXiv preprint arXiv:2004.02229*, 2020.

Zhang, Zhifei, S. Y. and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.