IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE IN CYBERSECURITY

Received May 9, 2019, accepted June 19, 2019, date of publication June 24, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924479

Latent-Space-Level Image Anonymization With Adversarial Protector Networks

TAEHOON KIM AND JIHOON YANG, (Member, IEEE)
Data Mining Research Laboratory, Department of Computer Science and Engineering, Sogang University, Seoul 121-742, South Korea

Corresponding author: Jihoon Yang (yangjh@sogang.ac.kr)

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) Grant funded by the Korea Government (MSIT) (A Development of Deidentification Technique Based on Differential Privacy) under Grant 2017-0-00498.

ABSTRACT Along with recent achievements in deep learning empowered by enormous amounts of training data, preserving the privacy of an individual related to the gathered data has been becoming an essential part of the public data collection and publication. Advancements in deep learning threaten traditional image anonymization techniques with model inversion attacks that try to reconstruct the original image from the anonymized image. In this paper, we propose a privacy-preserving adversarial protector network (PPAPNet) as an image anonymization tool to convert an image into another synthetic image that is both realistic and immune to model inversion attacks. Our experiments on various datasets show that PPAPNet can effectively convert a sensitive image into a high-quality and attack-immune synthetic image.

INDEX TERMS Adversarial learning, data privacy, deep learning, differential privacy, generative adversarial networks, machine learning, model inversion attacks.

I. INTRODUCTION

Stimulated by recent achievements in deep learning in different research domains such as video recommendation [9], facial recognition [36], and medical diagnosis [15], [39], [43], many companies and researchers are interested in using their own data to train state-of-the-art machine learning models. Well-known benchmark datasets [10], [25], [28], [41] might be enough for researchers to compare the performance of their model with others, but this cannot lead to applications of their model in the real world. Because there is no free lunch, it is essential for companies to re-train ML models using their own dataset for the best performance before applying it to commercial services. This is where privacy issues come into effect.

Datasets including collections of images, speech, or videos from millions of individuals are ripe with privacy risks. Chaudhuri *et al.* [6] states that simply releasing only statistics or pre-trained machine learning models on sensitive datasets may not be sufficient to preserve privacy. They propose *objective perturbation* as a privacy-preserving machine learning algorithm design based on the *sensitivity method* proposed by Dwork *et al.* [12]. Their algorithm is private under the ϵ-differential privacy definition defined by Dwork *et al.*

Synthetic data generation [4], [27], [42] is another technique in which sensitive data is partially or fully replaced with synthetic data before it is published. Synthetic data generation has taken the focus in recent years as a fundamental solution for privacy-preserving data publication. Beaulieu-Jones *et al.* [4] apply an objective perturbation [6] on ACGAN [33] to generate shareable biomedical data. However, the idea of applying objective perturbations to generative adversarial networks (GAN) [2], [5], [17], [33], [34], [44] while training the network with image datasets [25], [26], [28], [41] easily led to mode collapse [17], [34] in our prior attempts.

Instead of doing the hard work of trying to apply objective perturbations on GAN to generate a synthetic image in a differentially private way, we developed an advanced mechanism for traditional image anonymization, *adding noise to an image*. Recent research by Fredrikson *et al.* [16] suggests that it is possible to recover (up to a certain degree) individual faces from images that are blurred-out to protect anonymity using model inversion attacks. Our approach gives a solution to this problem by randomly manipulating features of an image rather than its pixels. In this work, we propose a *privacy-preserving adversarial protector network* (PPAPNet) as a tool to anonymize an image at the latent space level to simultaneously provide privacy and utility. PPAPNet consists of three networks: *protector, critic, and attacker*, as shown in Figure 1.

Latent space representation of an image is a vector that contains important features of an image, such as hair, skin color, and facial expression. Convolutional autoencoders [19], [37]

The associate editor coordinating the review of this manuscript and approving it for publication was Nan Cheng.

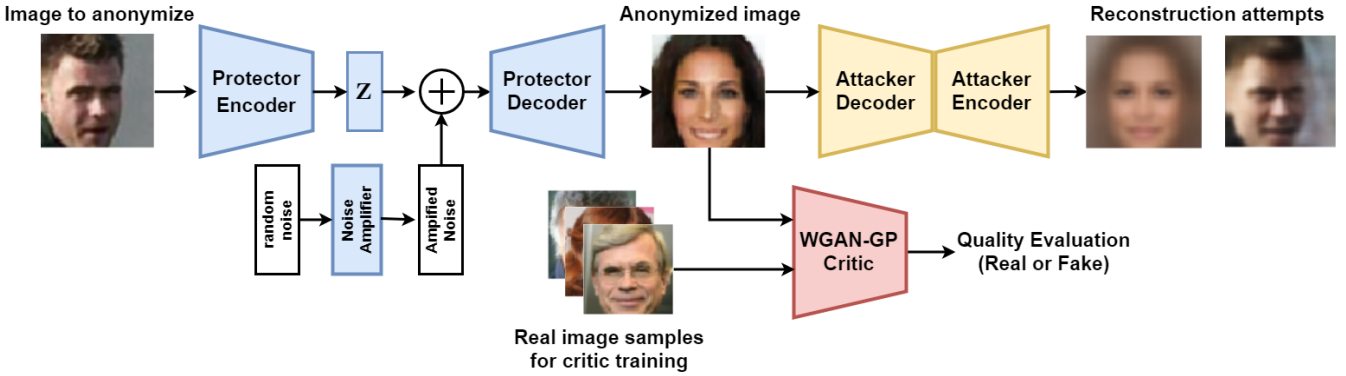
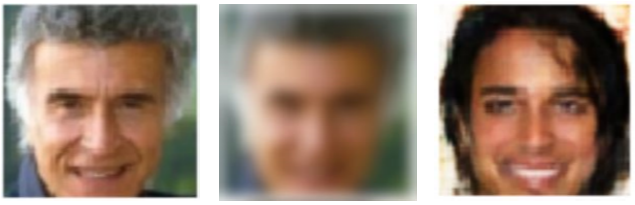


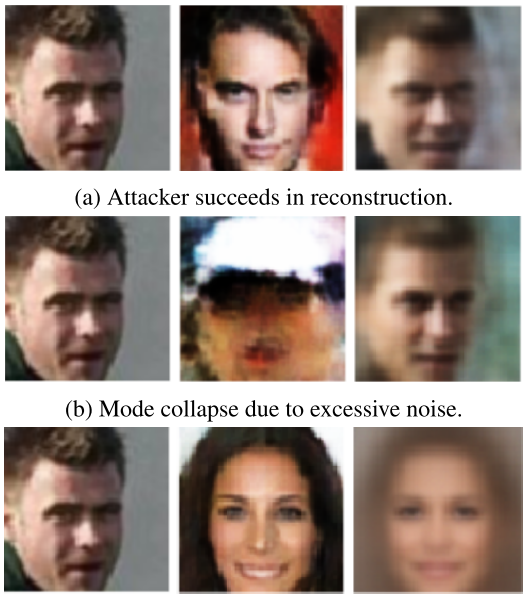
FIGURE 1. Proposed PPAPNet. The protector attempts to anonymize the input into a new image that is realistic and immune to privacy attacks.



(a) Original image. (b) Blurred image. (c) Our method.

FIGURE 2. Comparison between image blurred at the pixel level and image anonymized using PPAPNet on CelebA.

are widely used to extract this representation. While adding noise to a pixel just removes details of an image, we can directly modify important features of an image by manipulating its vector representation. Radford *et al.* [34] demonstrate that it is possible to change features of an image by applying vector arithmetic to its latent space representation. Taking everything into consideration, protector is a encoder-decoder network that encodes an image, Figure 2a, into its vector representation (z), manipulates the z vector, and decodes the manipulated z into a new anonymized image (Figure 2c). To train and make the anonymized image more realistic, additional mechanisms are required. Similar to Mariani *et al.* [29], we first initialize protector with a pre-trained convolutional autoencoder to start the training from a more stable point. Then, we adversarially train the protector with a WGAN-GP [18] critic. The WGAN-GP critic guides the protector to generate a realistic image by evaluating its quality. To ensure that the anonymized output of a PPAPNet is safe from model inversion attacks [16], we also add another encoder-decoder network, *attacker*, that tries to reconstruct the original, Figure 2a, from the anonymized image, as shown in Figure 2c. During the training process of a PPAPNet, protector defends from an attacker's inversion attack by adding noise to the z vector. However, simply adding random noise with various scales could allow attacker to successfully reconstruct the original image, as shown in Figure 3a. In addition, this could even lead protector to a mode collapse, as shown in Figure 3b. PPAPNet has a noise amplifier inside protector that learns optimal noise scaling parameters for each dimension of the z vector. A noise amplifier helps protector create images that are both realistic



(a) Attacker succeeds in reconstruction.

(b) Mode collapse due to excessive noise.

(c) Attacker fails in reconstruction.

FIGURE 3. Original image (left), anonymized image from PPAPNet (middle), and reconstructed image from attacker (right) when trained with CelebA.

and immune to privacy attacks, as shown in Figure 3c. The main contributions of our work are as follows:

- We propose an image anonymization deep neural network, PPAPNet, that transforms an image into another synthetic image by adding optimized noise to the original image's latent space representation.
- We introduce an overall training strategy that enables PPAPNet to generate a realistic image that cannot be distinguished from real images in the same domain and that protects the sensitive image from privacy attacks.
- We evaluate the proposed PPAPNet methodology with various image datasets.

Experimental results empirically demonstrate that our proposed PPAPNet leads to a higher level of image anonymization.

II. BACKGROUND

In this section, we briefly introduce important concepts of model inversion attacks, generative adversarial networks,

and differential privacy mechanisms. Although our work is not differential privacy guaranteed, we utilize the noise generation concept of Laplace and the Gaussian mechanism [12]–[14] in our noise amplifier.

A. MODEL INVERSION ATTACK

Fredrikson *et al.* [16] explore privacy issues in modern machine learning APIs, showing that confidential information can be exploited by adversarial clients in order to mount model inversion attacks. They also provide model inversion algorithms that can be used to infer sensitive features from decision trees or to extract images of training subjects from face recognition models.

The deblurring attack [16] is a type of model inversion (MI) attack that can reconstruct the original image from the blurred image. Let us assume an adversary has an image containing a blurred-out face and wishes to learn the identity of the corresponding individual. The adversary uses the blurred image as side information in a series of MI attacks to recover the original image up to a certain degree such that it can be classified correctly with face recognition models. This means that simply blurring an image at the pixel-level is vulnerable to MI attacks. In this paper, we mainly focus on the deblurring attack and propose an alternative way to anonymize an image that is different from simply blurring and noising.

B. GENERATIVE ADVERSARIAL NETWORKS

In recent years, generative adversarial networks (GANs) [2], [5], [17], [33], [34], [44] have been used as powerful tools to generate realistic images. The underlying idea is to train a generator and a discriminator in an adversarial mode. This idea is the most powerful concept so far among generative models. GAN is a two-player minimax game between a generator and a discriminator. From another perspective, GAN tries to minimize a distance or divergence between the model distribution (P_θ) and the real distribution (P_r). Nowozin *et al.* [32] states that any f-divergence can be used as the objective function. Using an appropriate f-divergence prevents mode collapse, which is a well-known problem when GAN's generator only draws one or a few foolish examples. Wasserstein GAN [2] and its improved version WGAN-GP [18] use the Earth Mover (EM) distance for the objective function and achieve state-of-the-art performance.

GANs are also widely used in the area of style transfer [7], [22], [23], [45]. Kim *et al.* [23] use deep convolutional encoder-decoder networks as generators to find mappings between two different image domains and a DCGAN [34] discriminator to evaluate the quality of mapped images. Parts of our work are focused on finding a certain mapping that can convert an image to another image in the same domain. Similar to Kim *et al.* [23], we use deep convolutional encoder-decoder networks to find this mapping. For better performance, we use a WGAN-GP critic (discriminator) to stabilize the training process. In Section 4, we also show that simply using this mapping to anonymize an image makes it vulnerable to an MI attack that tries to find the mapping between the original and the anonymized image using denoising autoencoders. We thus emphasize the importance of latent-space-level anonymization with a noise amplifier.

C. DIFFERENTIAL PRIVACY

Differential privacy was first introduced by Dwork *et al.* [12]–[14] and has been a strong standard for privacy guarantees for algorithms on aggregate databases. According to Dwork *et al.* [12], differential privacy intuitively captures the increased risk to one's privacy incurred by participating in a database. It was originally defined for two adjacent datasets that differ by a single element:

Definition 1: A randomized mechanism, $M : D \to R$, with domain D and range R satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs $d, d' \in D$ and for any subset of outputs $S \subseteq R$, it holds that:

$$\Pr[M(d) \in S] \leq e^{\epsilon} \Pr[M(d') \in S] + \delta \tag{1}$$

Laplace and Gaussian noise mechanisms are commonly used to approximate a deterministic real-valued function, $f : D \to R$, via additive noise calibrated to f's sensitivity s_f, which is defined as the maximum of the absolute distance $|f(d) - f(d')|$, where d and d' are adjacent inputs.

The Laplace noise mechanism is defined by:

$$M(d) \triangleq f(d) + Lap(0, b) \tag{2}$$

where $Lap(0, b)$ is the Laplace distribution with location 0 and scale b. This satisfies ϵ-differential privacy if b is $\frac{s_f}{\epsilon}$ [12].

The Gaussian noise mechanism is defined by:

$$M(d) \triangleq f(d) + N(0, \sigma^2) \tag{3}$$

where $N(0, \sigma^2)$ is a normal (Gaussian) distribution with mean 0 and standard deviation σ. This satisfies (ϵ, δ)-differential privacy if σ is $\sqrt{2\log(1.25/\delta)}\ \frac{s_f}{\epsilon}$ [12].

An interesting part of the Laplace and Gaussian noise mechanisms is that we can easily manage the intensity of noise by changing the privacy budgets ϵ and δ. Since the location or mean is always set to 0, it is easy to amplify the random noise sampled from $Lap(0, 1)$ or $N(0, 1)$ to the noise sampled from $Lap(0, b)$ or $N(0, \sigma^2)$. This is the key concept of the noise amplifier inside the proposed PPAPNet.

III. PPAPNET

Recent advances in deep learning threaten traditional image anonymization techniques with deep learning based MI attacks that try to reconstruct the original image from the anonymized image. The proposed PPAPNet methodology aims to convert an image into another synthetic image that is both realistic and immune to MI attacks. The model structure is depicted in Figure 1. PPAPNet consists of the protector P, the attacker A, and the critic C. P obtains an image x and anonymizes it into a new image $\tilde{x}$. A tries to reconstruct x from $\tilde{x}$. C evaluates the quality of $\tilde{x}$.

Algorithm 1 PPAPNet Training With Default Values of $\lambda = 10$, $n_{critic} = 5$, $n_{attacker} = 1$, $\alpha = 0.0001$, $\beta_1 = 0.5$, and $\beta_2 = 0.9$.

Require: Initial protector parameters, θ_0, initial critic parameters, wc_0, and initial attacker parameters, wa_0.
Require: Number of critic iterations and attacker iterations per protector iteration, n_{critic} and $n_{attacker}$.
Require: Batch size, m, gradient penalty coefficient, λ, Adam hyperparameters, α, β_1, and β_2.

while θ has not converged **do**
  for $t = 1, \ldots, n_{critic}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample real data sets $x \sim \mathbb{P}_r$ and $x' \sim \mathbb{P}_r$ and a random number $\epsilon \sim U[0, 1]$.
      $\tilde{x} \leftarrow P_\theta(x)$
      $\hat{x} \leftarrow \epsilon x' + (1 - \epsilon)\tilde{x}$
      $L_c^{(i)} \leftarrow C_{wc}(\tilde{x}) - C_{wc}(x') + \lambda(\|\nabla_{\hat{x}} C(\hat{x})\|_2 - 1)^2$
    $wc \leftarrow Adam(\nabla \frac{1}{m} \sum_{i=1}^{m} L_c^{(i)}, wc, \alpha, \beta_1, \beta_2)$
  for $t = 1, \ldots, n_{attack}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample real data set $x \sim \mathbb{P}_r$.
      $\tilde{x} \leftarrow P_\theta(x)$
      $L_a^{(i)} \leftarrow \|x - A_{wa}(\tilde{x})\|_2$
    $wa \leftarrow Adam(\nabla \frac{1}{m} \sum_{i=1}^{m} L_a^{(i)}, wa, \alpha, \beta_1, \beta_2)$
  for $i = 1, \ldots, m$ **do**
    Sample real data set $x \sim \mathbb{P}_r$.
    $\tilde{x} \leftarrow P_\theta(x)$
    $L_p^{(i)} \leftarrow -C_{wc}(\tilde{x}) - \|x - A_{wa}(\tilde{x})\|_2$
  $\theta \leftarrow Adam(\nabla \frac{1}{m} \sum_{i=1}^{m} L_p^{(i)}, \theta, \alpha, \beta_1, \beta_2)$

A. MODEL ARCHITECTURE

The protector P and the attacker A take an image of size $n \times n \times k$ and feed it through an encoder-decoder pair. The encoder parts of P and A are composed of 5 convolution layers with 5×5 and a stride size of 2, each followed by a batch normalization [21] and a leaky ReLU [40]. The decoder part is composed of 5 deconvolution (transposed convolution) [31] layers with 5×5 and a stride size of 2, each followed by a batch normalization and ReLU [30]. For the last activation function of the decoder, we used a sigmoid instead of ReLU to set the final image output range between [0, 1]. The protector network also has an additive noise layer and a noise amplifier between the encoder and the decoder. The final output of the encoder is connected to a fully-connected layer and reduced to a 128-dimensional vector z. The noise amplifier adds noise to z and projects it to the same size as the decoder input.

The critic C takes an image of size $n \times n \times k$ and decides whether it is real or fake. C is composed of 4 or 5 convolution layers with 5×5 and a stride size of 2, each followed by a layer normalization [3] and a leaky ReLU. We use 5 convolution layers for an image of size $64 \times 64 \times 3$ and 4 for other images. The discriminator has an additional fully connected layer to follow the Wasserstein distance metric.

B. ADVERSARIAL TRAINING

To analyze the performance of A, we define a new term, *privacy loss* L_{priv}, as the reconstruction loss between x and $\tilde{x}$. We apply the l_2 loss for the reconstruction loss of L_{priv}.

$$L_{priv} = \|x - \tilde{x}\|_2 \tag{4}$$

The critic C attempts to label x' as real and $\tilde{x}$ as fake. Although x and x' are both sampled from the real data distribution, we distinguish x and x' for better understanding of our critic loss L_c. For better convergence of L_c, we use WGAN-GP's critic loss [18] instead of the original GAN's discriminator loss [17]. All considered, our final critic loss becomes:

$$L_c = \mathop{\mathbb{E}}_{\tilde{x} \sim \mathbb{P}_p}[C(\tilde{x})] - \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_r}[C(x')] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} C(\hat{x})\|_2 - 1)^2] \tag{5}$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_p$ is the model distribution implicitly defined by $\tilde{x} = P(x)$. Gulrajani *et al.* [18] apply a gradient penalty to the WGAN critic with $\hat{x} = \epsilon x' + (1 - \epsilon)\tilde{x}$ where ϵ is a random sample from $\mathbb{U}[0, 1]$. We apply a gradient penalty on C with $\lambda = 10$.

The protector P has to return an image that maximizes $C(\tilde{x})$ and L_{priv}.

$$L_P = -\mathop{\mathbb{E}}_{\tilde{x} \sim \mathbb{P}_p}[C(\tilde{x})] - L_{priv} \tag{6}$$

The attacker A just tries to minimize L_{priv}.

$$L_A = L_{priv} \tag{7}$$

When the PPAPNet modules are initialized, all the weights in the protector, attacker, and critic are fine-tuned by carrying out an adversarial training in Algorithm 1. When the protector parameter θ converges, P is optimized to convert an image into a synthetic image that is realistic enough to fool the

TABLE 1. Noise amplifier parameters used in our experiments. PPAP-None is a model without any additive noise mechanism. *Normal* stands for a normal (Gaussian) distribution and *Laplace* stands for a Laplace distribution. s_e is the approximate sensitivity of the protector's encoder output.

Model	Noise sampling distribution	Noise amplifier function (σ)	Modified vector ($\tilde{z_d}$)
PPAP-None	None	None	z_d
PPAP-$tanh$-(ϵ)	$n_0 \sim Laplace(0,1)$	$\sigma(2;\epsilon) = \frac{2}{\epsilon}$	$z_d + \sigma(2;\epsilon)n_0$
PPAP-$tanh$-(ϵ,δ)	$n_0 \sim Normal(0,1)$	$\sigma(2;\epsilon,\delta) = \sqrt{2\log(1.25/\delta)}\ \frac{2}{\epsilon}$	$z_d + \sigma(2;\epsilon,\delta)n_0$
PPAP-s_e-(ϵ)	$n_0 \sim Laplace(0,1)$	$\sigma(s_e;\epsilon) = \frac{s_e}{\epsilon}$	$z_d + \sigma(s_e;\epsilon)n_0$
PPAP-s_e-(ϵ,δ)	$n_0 \sim Normal(0,1)$	$\sigma(s_e;\epsilon,\delta) = \sqrt{2\log(1.25/\delta)}\ \frac{s_e}{\epsilon}$	$z_d + \sigma(s_e;\epsilon,\delta)n_0$

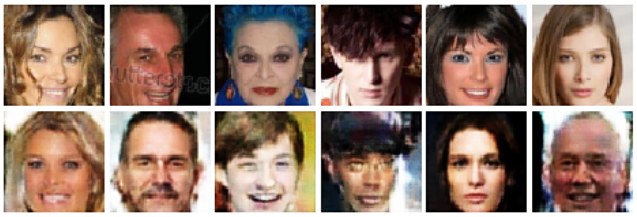


FIGURE 4. Samples from PPAPNet trained on CelebA. PPAPNet anonymizes an image (top) into a new image (bottom).

critic C and strong enough to defend against model inversion attacks by the attacker A.

C. NOISE AMPLIFIER

The noise amplifier Σ is placed in between the encoder E_P and the decoder D_P of the protector. E_P encodes an image in its latent space representation in the form of a d-dimensional vector z_d. For each dimension of z_d, Σ tries to find the optimal scale factor σ^* for random noise n sampled from a probability distribution. For σ^*, Σ manipulates z_d forming $\tilde{z_d}$ where $\tilde{z_d} = z_d + \sigma^* n$ and returns $\tilde{z_d}$ to D_P. Σ is a neural network that approximates σ^* with a noise amplifier function σ and its d-dimensional weight parameters. As we mentioned in Section 2, we utilized the Laplace and Gaussian noise mechanisms to design our noise amplifier. We can make noise sampled from $Lap(0, 1)$ to follow $Lap(0, b)$ by multiplying the scale factor b by the original noise. Noise sampled from $N(0, 1)$ also follows $N(0, \sigma)$ if it is multiplied by σ. In our experiment, we mainly use two types of $\Sigma, \sigma(s_e; \epsilon) = \frac{s_e}{\epsilon}$ with a weight parameter of ϵ and $\sigma(s_e; \epsilon, \delta) = \sqrt{2\log(1.25/\delta)}\ \frac{s_e}{\epsilon}$ with weight parameters ϵ and δ where s_e is the approximate sensitivity of E_P.

It is impossible to find the exact sensitivity of the unknown E_P during the training. Instead, we initialize the encoder E_P and decoder D_P with a pre-trained autoencoder using all the images in the training set. In this work, we apply the l_2 loss minimization for the autoencoder training. After initialization, we freeze all the layers of E_P so that the weights of E_P will not change during the training. The approximate sensitivity s_e is defined by:

$$s_e = \max_{x_i \sim \mathbb{S}_t} E_P(x_i) - \min_{x_j \sim \mathbb{S}_t} E_P(x_j) \tag{8}$$

where x_i and x_j are images sampled from the training set, $\mathbb{S}_t$. Another way to cope with the sensitivity issue is to normalize z in the range $[-1, 1]$ using $tanh$. This allows us to assume that the upper bound of E_P is 2 and use it as s_e. In Section 4, we evaluate two sensitivity approximation techniques and their privacy-preserving performances.

Now, the ith value of the modified vector $\tilde{z_i}$ becomes:

$$\tilde{z_i} = z_i + \sigma(s_e)n_0 \tag{9}$$

where the z_i is the ith value of the original vector z_d, n_0 is the initial noise, s_e is the approximate sensitivity, and σ is the noise amplifier function. The values of ϵ and δ are initialized to 1 and 1e−8, respectively, and optimized throughout the training using gradient descent in the direction of maximizing L_{priv}.

TABLE 2. Detailed information about datasets.

Dataset Name	Resolution	Training set	Test set
MNIST	$28 \times 28 \times 1$	60,000	10,000
CIFAR-10	$32 \times 32 \times 3$	50,000	10,000
CelebA	$64 \times 64 \times 3$	200,000	2,599
LSUN Bedroom	$64 \times 64 \times 3$	3,030,000	3,342

TABLE 3. Inception scores on unsupervised CIFAR-10.

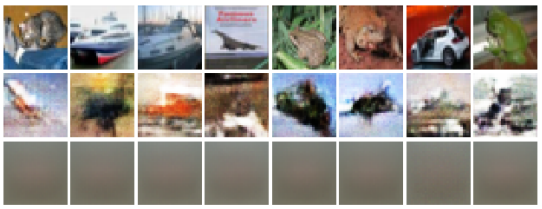
Method	Score
ALI [11] (in [38])	$5.34 \pm .05$
BEGAN [5]	5.62
DCGAN [34] (in [20])	$6.16 \pm .07$
Improved GAN (-L+HA) [35]	$6.86 \pm .06$
EGAN-Ent-VI [8]	$7.07 \pm .10$
DFM [38]	$7.72 \pm .13$
WGAN-GP Resnet [18]	$7.86 \pm .07$
PPAP-s_e-(ϵ) (ours)	$2.83 \pm .01$
PPAP-s_e-(ϵ,δ) (ours)	$2.60 \pm .01$

D. IMPLEMENTATION DETAILS

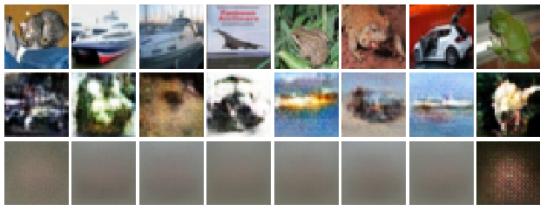
Our implementation is done using TensorFlow [1]. We trained our network with the Adam optimizer [24] with a learning rate of 0.0001, $\beta_1 = 0.5$, and $\beta_2 = 0.9$. We trained our model using a single NVIDIA Titan V GPU with a mini-batch size of 256. In our experiments, we use models with 5 different types of initial noise n_0 and noise amplifier function σ listed in Table 1. For CelebA and LSUN bedroom, we center-cropped and resized images to $64 \times 64 \times 3$. More details regarding each dataset are shown in Table 2. We only use test images for fair evaluations and demonstrations.

IV. EXPERIMENTS

We experimentally demonstrate our model's performance using the MNIST [26], the CIFAR-10 [25], the CelebA [28], and the LSUN [41] bedroom datasets. We first evaluate the sample diversity of PPAPNet with an Inception score [35]



(a) PPAP-s_e-(ϵ) CIFAR-10.



(b) PPAP-s_e-(ϵ, δ) CIFAR-10.

FIGURE 5. Samples from PPAPNets trained on CIFAR-10. Original images (top), anonymized samples (middle), and reconstruction results (bottom) of the adversary are shown in each subfigure.

on unsupervised CIFAR-10. We also show the importance of the noise amplifier in PPAPNet by comparing the privacy gain with different experimental setups. Sample images anonymized with PPAPNet are provided in Figure 7.

A. SAMPLE DIVERSITY ON UNSUPERVISED CIFAR-10

We measure the Inception score of PPAPNet trained with unsupervised CIFAR-10 and compare with other published GAN models. In Table 3, the Inception scores of our models are much lower than those of all other published models. Low Inception scores usually mean higher rates of mode collapse. However, we do not find significant instances of mode collapse in our models. Samples from PPAPNets trained on CIFAR-10 are shown in Figure 5. Considering the fact that our basic model structure is similar to DCGAN [34] and WGAN-GP [18], we find the main cause of this phenomenon is our latent-space-level anonymization mechanism. Since we use an encoder that is pre-trained with an autoencoder to extract important features of an image, the output vector z of the protector's encoder contains meaningful features that can distinguish an image from others in the same domain. However, the main objective of the anonymization is to manipulate distinct features of an image to protect the privacy of an individual from adversarial attacks. PPAPNet accomplishes this objective by making the original feature indistinguishable from others with its differential privacy (DP) oriented additive noise mechanism. This makes the Inception network work harder to correctly classify the anonymized image lowering the Inception score.

B. PRIVACY GAIN

To evaluate the privacy preserving performance of different noise amplifiers, we use a denoising autoencoder as an adversary to perform attacks on the PPAPNets shown in Table 1. The adversary tries to reconstruct the original image from the PPAPNet-anonymized image. The term *privacy gain* is the l_2 loss between the original image and the reconstructed image. Higher privacy gain means better anonymization performance. In our experiment, we use the attacker A's model architecture for the adversary and train it using the MNIST, CelebA, and LSUN bedroom datasets. After training the adversary with training images and its PPAPNet-anonymized version, we measure the privacy gain of each model in Table 1 with test images. The results are shown in Tables 4a, 4b, and 4c. Models with sensitivity approximation mechanisms (8) perform best in most of the cases with the exception of MNIST. The difference between PPAPNet-None and other models show that the noise amplifier module is an essential part of the PPAPNet architecture.

We also train the adversary with blurred CelebA images to visualize the degree of privacy gain. We train the adversary on different sets of images blurred with a Gaussian filter with different standard deviations (σ) ranging from 1 to 15. Evaluated with the trained adversary and blurred test images, the Gaussian filter with $\sigma = 15$ shows 1.24×10^{-2} of privacy gain, while the maximum privacy gain of PPAPNet trained on CelebA is 5.53×10^{-2} (PPAP-s_e-(ϵ) in Table 4b). More detailed relationships between values of σ and their corresponding privacy gains are demonstrated in Figure 6. The overall results state that simply blurring an image at the pixel-level is vulnerable to reconstruction attacks with denoising autoencoders and that our proposed PPAPNet is a solution.

C. ANONYMIZED SAMPLES

We present the anonymized image samples along with their original and the reconstruction attack result of an adversary (explained in Section 4.2) in Figure 7. Although every PPAPNet model successfully generates realistic outputs, their performances on privacy preservation are quite different. The Adversary trained against PPAP-None easily finds the internal mapping of PPAP-None and successfully reconstructs the

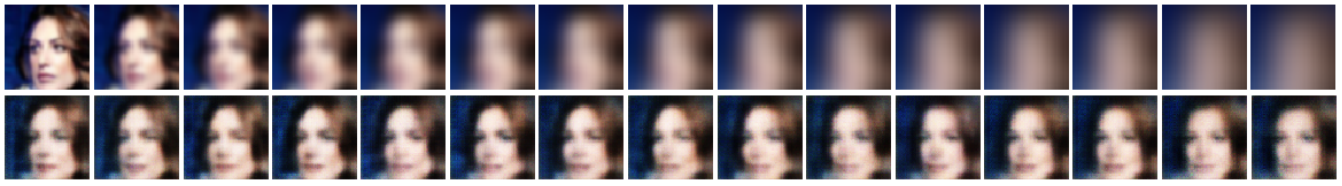


FIGURE 6. Blurred images with Gaussian filters (top) and images reconstructed by the adversary (bottom). The σ of the Gaussian filter ranges from 1 to 15 (left to right). The minimum amount of privacy gain is 6.26×10^{-3} ($\sigma = 1$) and the maximum is 1.24×10^{-2} ($\sigma = 15$).

(a) PPAP-None MNIST. (b) PPAP-$tanh$-(ϵ) MNIST. (c) PPAP-$tanh$-(ϵ, δ) MNIST.

(d) PPAP-None CelebA. (e) PPAP-s_e-(ϵ) CelebA. (f) PPAP-s_e-(ϵ, δ) CelebA.

(g) PPAP-None LSUN bedroom. (h) PPAP-s_e-(ϵ) LSUN bedroom. (i) PPAP-s_e-(ϵ, δ) LSUN bedroom.

FIGURE 7. **Samples from PPAPNets trained on various datasets. Original images (top), anonymized samples (middle), and reconstruction results (bottom) of the adversary are shown in each subfigure.**

TABLE 4. **Privacy gain on different datasets.**

(a) MNIST.

Method	Privacy gain
PPAP-None	7.78×10^{-3}
PPAP-$tanh$-(ϵ)	8.54×10^{-2}
PPAP-$tanh$-(ϵ, δ)	9.98×10^{-2}
PPAP-s_e-(ϵ)	6.75×10^{-2}
PPAP-s_e-(ϵ, δ)	5.64×10^{-2}

(b) CelebA.

Method	Privacy gain
PPAP-None	5.83×10^{-3}
PPAP-$tanh$-(ϵ)	$2.03e \times 10^{-2}$
PPAP-$tanh$-(ϵ, δ)	1.63×10^{-2}
PPAP-s_e-(ϵ)	5.5×10^{-2}
PPAP-s_e-(ϵ, δ)	5.35×10^{-2}

(c) LSUN Bedroom.

Method	Privacy gain
PPAP-None	1.26×10^{-2}
PPAP-$tanh$-(ϵ)	3.15×10^{-2}
PPAP-$tanh$-(ϵ, δ)	3.97×10^{-2}
PPAP-s_e-(ϵ)	5.83×10^{-2}
PPAP-s_e-(ϵ, δ)	5.51×10^{-2}

original image, while the adversary trained against PPAPNet with the noise amplifier only omits similar meaningless images. Even if the adversary finds the internal mapping and reconstructs the modified feature vector $\tilde{z}$ it fails to extract the real feature vector z from $\tilde{z}$.

V. CONCLUSION

In this work, we present a methodology to anonymize the latent space representation of an image. In the proposed PPAPNet framework, the protector and the attacker are trained in adversarial mode to find the best way to protect the image from possible privacy risks. The noise amplifier inside the protector plays an important role in noise optimization for effective image anonymization. We evaluate the proposed PPAPNet with different metrics and datasets to demonstrate its powerful performance. In the future, we hope to apply the latent-space-level anonymization methodology to a broader range of data domains including video, text, and speech.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat, (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, D. Precup and Y. W. Teh, Eds. Sydney, NSW, Australia: International Convention Centre, vol. 70, Aug. 2017, pp. 214–223.

[3] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, Jul. 2016.

[4] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, and C. S. Greene, "Privacy-preserving generative deep neural networks support clinical data sharing," *BioRxiv*, 2017.

[5] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," *CoRR*, Mar. 2017.

[6] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," *CoRR*, Nov. 2017.

[8] Z. Dai, A. Almahairi, P. Bachman, E. Hovy, and A. Courville, "Calibrating energy-based generative adversarial networks," *CoRR*, Feb. 2017.

[9] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The YouTube video recommendation system," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, New York, NY, USA, 2010, pp. 293–296.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[11] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially learned inference," 2017, *arXiv:1606.00704*. [Online]. Available: https://arxiv.org/abs/1606.00704

[12] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Germany: Springer, 2006, pp. 1–12.

[13] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*. Berlin, Germany: Springer, 2008, pp. 1–19.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Germany: Springer, 2006, pp. 265–284.

[15] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, Jan. 2017.

[16] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2015, pp. 1322–1333.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2014, pp. 2672–2680.

[18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," *CoRR*, Dec. 2017.

[19] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Eds. Cham, Switzerland: Springer, 2017, pp. 373–382.

[20] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," *CoRR*, Dec. 2016.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 448–456.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, Nov. 2016.

[23] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," Mar. 2017, *arXiv:1703.05192*. [Online]. Available: https://arxiv.org/abs/1703.05192

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, Dec. 2014.

[25] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10," Can. Inst. Adv. Res., Toronto, ON, Canada, Tech, Rep.

[26] Y. LeCun and C. Cortes, "MNIST handwritten digit database," Tech. Rep., 2010.

[27] H. Li, L. Xiong, L. Zhang, and X. Jiang, "DPSynthesizer: Differentially private data synthesizer for privacy preserving data sharing," *Proc. VLDB Endowment*, vol. 7, no. 13, pp. 1677–1680, 2014.

[28] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 3730–3738.

[29] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," *CoRR*, Mar. 2018.

[30] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2010, pp. 807–814.

[31] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *CoRR*, May 2015.

[32] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 271–279.

[33] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," Oct. 2016, *arXiv:1610.09585*. [Online]. Available: https://arxiv.org/abs/1610.09585

[34] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, Nov. 2015.

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *CoRR*, Jun. 2016.

[36] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.

[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[38] D. Warde-Farley and Y. Bengio, "Improving generative adversarial networks with denoising feature matching," in *Proc. ICLR*, 2017, pp. 1–11.

[39] Z. Xiao, R. Huang, Y. Ding, T. Lan, R. Dong, Z. Qin, X. Zhang, and W. Wang, "A deep learning-based segmentation method for brain tumor in MR images," in *Proc. IEEE 6th Int. Conf. Comput. Adv. Bio Med. Sci. (ICCABS)*, Oct. 2016, pp. 1–6.

[40] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, May 2015.

[41] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," *CoRR*, Jun. 2015.

[42] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via Bayesian networks," *ACM Trans. Database Syst.*, vol. 42, no. 4, pp. 25:1–25:41, Oct. 2017.

[43] Q. Zhang, Y. Xiao, J. Suo, J. Shi, J. Yu, Y. Guo, Y. Wang, and H. Zheng, "Sonoelastomics for breast tumor classification: A radiomics approach with clustering-based feature selection on sonoelastography," *Ultrasound Med. Biol.*, vol. 43, pp. 1058–1069, May 2017.

[44] J. J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *CoRR*, Sep. 2016.

[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, Mar. 2017.

TAEHOON KIM received the B.E. degree in computer science and engineering from Sogang University, in 2018, where he is currently pursuing the Ph.D. degree in computer science and engineering. His research interests include machine learning, artificial intelligence, knowledge discovery and data mining, learning representations, text/web mining, text classification, text summarization, information retrieval, neural and evolutionary computation, and pattern recognition.

JIHOON YANG received the B.E. degree in computer science from Sogang University, in 1987, and the M.E. and Ph.D. degrees in computer science from Iowa State University, in 1989 and 1999, respectively. From 1999 to 2000, he was a Research Staff Member with HRL Laboratories, LLC. From 2000 to 2002, he was a Professional Staff Member with SRA International, Inc. Since 2002, he has been a Faculty Member with the Department of Computer Science and Engineering, Sogang University. His research interests include machine learning, artificial intelligence, knowledge discovery and data mining, bioinformatics and computational biology, text/web mining, text classification, text summarization, information retrieval, neural and evolutionary computation, and pattern recognition.