Leveraging Contaminated Datasets to Learn Clean-Data Distribution with Purified Generative Adversarial Networks

Bowen Tian[1], **Qinliang Su**[1,2 *], **Jianxing Yu**[3]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[3] School of Artificial Intelligence, Sun Yat-sen University, Guangdong, China
tianbw@mail2.sysu.edu.cn, {suqliang,yujx26}@mail.sysu.edu.cn

Abstract

Generative adversarial networks (GANs) are known for their strong abilities on capturing the underlying distribution of training instances. Since the seminal work of GAN, many variants of GAN have been proposed. However, existing GANs are almost established on the assumption that the training dataset is clean. But in many real-world applications, this may not hold, that is, the training dataset may be contaminated by a proportion of undesired instances. When training on such datasets, existing GANs will learn a mixture distribution of desired and contaminated instances, rather than the desired distribution of desired data only (target distribution). To learn the target distribution from contaminated datasets, two purified generative adversarial networks (PuriGAN) are developed, in which the discriminators are augmented with the capability to distinguish between target and contaminated instances by leveraging an extra dataset solely composed of contamination instances. We prove that under some mild conditions, the proposed PuriGANs are guaranteed to converge to the distribution of desired instances. Experimental results on several datasets demonstrate that the proposed PuriGANs are able to generate much better images from the desired distribution than comparable baselines when trained on contaminated datasets. In addition, we also demonstrate the usefulness of PuriGAN on downstream applications by applying it to the tasks of semi-supervised anomaly detection on contaminated datasets and PU-learning. Experimental results show that PuriGAN is able to deliver the best performance over comparable baselines on both tasks.

Introduction

Learning data distribution from a dataset can be applied to various kinds of applications, like inpainting (Yu et al. 2018; Liu et al. 2021), anomaly detection (Schlegl et al. 2017; Zenati et al. 2018; Akcay, Atapour-Abarghouei, and Breckon 2018), image translation (Isola et al. 2017; Liu, Breuel, and Kautz 2017), AI medical diagnosis (Kazeminia et al. 2020; Izadi et al. 2018), *etc.* Among the existing methods of distribution learning, generative adversarial networks (GANs) (Goodfellow et al. 2014) and variational auto-encoder (VAE) (Kingma and Welling 2014) are the two most widely used ones. However, existing deep generative models are mostly established on the assumption of clean training datasets, that is, all training instances are drawn from the target distribution that we are interested in. But in real-world applications, it is quite common to see that some instances from an undesired distribution is mistakenly put into the training dataset, resulting in a contaminated dataset. This could be caused by the lack of sufficient expertise or enough labor to correctly recognize every undesired instance when building the dataset, or the high cost to clean a very large contaminated dataset etc. For example, it has been reported that a tiny ratio of images in ImageNet were labelled with incorrect categories, although their impacts on the final model trained on it is negligible due to their small proportion (Northcutt, Jiang, and Chuang 2021). But for many applications, the quality of collected datasets could be much poorer than ImageNet. When training on such datasets, generative models will only capture the distribution of the entire dataset, that is, mixture of the target distribution and contamination distribution. However, as we use the generative models, our primary intention is always to learn the distribution of desired instances (*i.e.* target distribution), which can later be used to generate new desired instances or assist downstream tasks like anomaly detection (Schlegl et al. 2017; Zenati et al. 2018; Akcay, Atapour-Abarghouei, and Breckon 2018), inpainting (Yu et al. 2018; Liu et al. 2021) etc. Therefore, investigating how to learn a generative model that only captures the distribution of desired instances from a contaminated dataset is important both theoretically and practically.

Generally, the goal above cannot be achieved by solely leveraging the contaminated dataset, which is denoted by $\mathcal{X}$. In this paper, in addition to $\mathcal{X}$, we also assume the availability of another small dataset $\mathcal{X}^-$ that is only composed of contamination instances. In many real-world applications, it is often possible to collect a small number of representative contamination instances, although collecting a large number of them may be difficult. For example, in the task of anomaly detection, it is possible to collect some anomalies, which can be seen as the contamination instances here. In fact, several recent works have been proposed to introduce an extra negative dataset and leverage it to boost the generation performance (Asokan and Seelamantula 2020; Sinha et al. 2020). However, they are all restricted to the scenario that the dataset $\mathcal{X}$ is clean. To only learn the distribution

[*] Corresponding author.

of desired instances from a contaminated dataset, a possible solution is to adopt a two-stage training strategy: 1) first, seeking to separate out the target samples from $\mathcal{X}$; 2) then, training generative models on the separated target samples. The objective of first stage can be partially achieved by resorting to positive-unlabelled (PU) learning (Elkan and Noto 2008; Kiryo et al. 2017; Du Plessis, Niu, and Sugiyama 2015; Kato, Teshima, and Honda 2018; Bekker and Davis 2020), whose goal is to partition unlabelled instances into two classes by leveraging datasets $\mathcal{X}$ and $\mathcal{X}^-$. However, it is observed that it is generally difficult to obtain a satisfactory performance at the first stage, especially when the data instances are complex. The poor performance of the first stage will be passed down to the second stage, resulting in an even worse performance of the whole model. Moreover, the valuable negative dataset $\mathcal{X}^-$ is not fully utilized by the two-stage method since it is only used in the first stage to separate out desired instance but is never used in the second stage.

In this paper, two purified generative adversarial networks (PuriGAN) are proposed, which can not only be trained in an end-to-end manner by simultaneously leveraging the two datasets $\mathcal{X}$ and $\mathcal{X}^-$, but also are guaranteed to converge to the target distribution theoretically. This is achieved by augmenting the discriminator to have it able to distinguish between the target and contaminated instances, in addition to the basic role of discriminating real and generated ones. The augmented discriminator can prevent the generator from generating contaminated instances, while the generator can generate extra instances to increase the discrimination ability of discriminator. This further improves the discriminator's robustness even in the case of insufficient data. Extensive experiments are conducted to evaluate the target-data generation ability of PuriGAN when it is trained on contaminated datasets. Experimental results demonstrate that the proposed PuriGANs are able to generate much better desired images over competitive baselines under various kinds of conditions. In addition, we also apply PuriGAN to two downstream applications, semi-supervised anomaly detection on contaminated dataset and PU-learning, with the experimental results demonstrating that PuriGAN outperforms comparable baselines remarkably.

The Proposed Purified Generative Adversarial Networks

Problem Description

In this problem, we suppose the training dataset

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \tag{1}$$

is not pure, but is contaminated by some undesired instances. That is, the training dataset $\mathcal{X}$ contains desired and undesired (contamination) instances simultaneously. Mathematically, we can think the instances $\mathbf{x}_i$ are drawn from the mixture distribution

$$p_d(\mathbf{x}) = \pi p^+(\mathbf{x}) + (1-\pi)p^-(\mathbf{x}), \tag{2}$$

where $p^+(\mathbf{x})$ and $p^-(\mathbf{x})$ denote the distributions of desired instances (target distribution) and contamination instances (contamination distribution), respectively; and π is the proportion of desired instances . In addition to $\mathcal{X}$, there exists another training dataset

$$\mathcal{X}^- = \{\mathbf{x}_1^-, \mathbf{x}_2^-, \cdots, \mathbf{x}_m^-\}, \tag{3}$$

which is only composed of contamination instances with $\mathbf{x}_i^- \sim p^-(\mathbf{x})$. Here, we argue that it is often possible to obtain a small number of contamination instances. For examples, in anomaly detection, we can collect a small number of anomalies, which can be viewed as the contamination instances; or given a contaminated dataset, we can assign some labors to find a fraction of contaminations in the dataset manually. But due to the often low proportion/frequency of contamination instances, the number of collected contamination instances cannot be too large. Thus, the size of $\mathcal{X}^-$ is assumed to be much smaller than $\mathcal{X}$.

The problem concerned in this paper is to obtain a generative model which only captures the target distribution $p^+(\mathbf{x})$ by leveraging the available datasets $\mathcal{X}$ and $\mathcal{X}^-$. Obviously, if we directly train a generative model on the dataset $\mathcal{X}$, the generative distribution $p_g(\mathbf{x})$ will only converge to the distribution of dataset $\mathcal{X}$, *i.e.*, the mixture distribution $p_d(\mathbf{x}) = \pi p^+(\mathbf{x}) + (1-\pi)p^-(\mathbf{x})$, instead of our desired target distribution $p^+(\mathbf{x})$. Thus, the key to address this problem lies at how to leverage the provided contamination instances $\mathbf{x}_i^-$ effectively, in addition to the dataset $\mathcal{X}$.

PuriGAN with Two-Level Discriminator

To address the problem above, in this paper, we establish our model on the framework of GANs, or more specifically on the least-square GAN (LSGAN), thanks to its flexibility in the design of discriminators. In LSGAN, the generator $G(\cdot)$ and discriminator $D(\cdot)$ are updated as

$$\begin{aligned}\min_D V_{LS}(D) =& \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\left[(D(\mathbf{x})-1)^2\right] \\ &+ \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}\left[(D(\mathbf{x})-0)^2\right], \end{aligned} \tag{4}$$

$$\begin{aligned}\min_G V_{LS}(G) =& \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\left[(D(\mathbf{x})-0.5)^2\right] \\ &+ \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[(D(G(\mathbf{z}))-0.5)^2\right], \end{aligned} \tag{5}$$

where $p_g(\mathbf{x})$ denotes the distribution of generated samples, that is, $G(\mathbf{z}) \sim p_g(\mathbf{x})$; and $p(\mathbf{z})$ is a standard normal distribution. It can be seen from (4) and (5) that LSGAN encourages the discriminator to output 1 for samples from the data distribution $p_d(\mathbf{x})$ and 0 for the generated ones, while forcing the generator to confuse the discriminator, *i.e.*, letting it output 0.5. The training objective of LSGAN is consistent with traditional GANs, except that it is implemented under the least-squared loss. From (4), it can be seen that the optimal discriminator of LSGAN is $D^*(\mathbf{x}) = \frac{p_g(\mathbf{x})}{p_d(\mathbf{x})+p_g(\mathbf{x})}$, which views all samples from $p_d(x) = \pi p^+(\mathbf{x}) + (1-\pi)p^-(\mathbf{x})$ are identical and does not distinguish between the samples from $p^+(\mathbf{x})$ and $p^-(\mathbf{x})$.

To enable the LSGAN to learn the desired data distribution from contaminated datasets, we propose to augment the LSGAN's discriminator by adding an extra term $\mathbb{E}_{\mathbf{x}\sim p^-(\mathbf{x})}\big[(D(\mathbf{x})-0)^2\big]$ in the $V_{LS}(D)$ to enable the induced discriminator to recognize the contamination instances

$$\min_D V(D) = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\left[(D(\mathbf{x})-1)^2\right] + \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}\left[(D(\mathbf{x})-0)^2\right] + \lambda\mathbb{E}_{\mathbf{x}\sim p^-(\mathbf{x})}\left[(D(\mathbf{x})-0)^2\right], \tag{6}$$

where λ is a weighting parameter; and $p_g(\mathbf{x})$ denotes the distribution of generated samples. Obviously, with the extra term $\mathbb{E}_{\mathbf{x}\sim p^-(\mathbf{x})}\big[(D(\mathbf{x})-0)^2\big]$, the discriminator seeks to output 0 for samples from $\mathcal{X}^-$ and 1 for samples from $\mathcal{X}$. Although $\mathcal{X}$ contains both desired and contamination instances, if we set λ to be a very large value, the discriminator can recognize the contamination instances from $p^-(\mathbf{x})$. With the discriminator $D(\cdot)$ derived from (6), if we train the generator to have $D(\cdot)$ outputting a nonzero value c, the generator will endeavour to avoid generating instances that look like contamination instances. Specifically, we propose to update the generator as

$$\min_G V(G) = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\big[(D(\mathbf{x})-c)^2\big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[(D(G(\mathbf{z}))-c)^2\big] + \mathbb{E}_{\mathbf{x}\sim p^-(\mathbf{x})}\big[(D(\mathbf{x})-c)^2\big], \tag{7}$$

where c could be any value within $(0,1)$. With the updating rules specified in (6) and (7) for $D(\cdot)$ and $G(\cdot)$, we can prove that under some conditions, the generator distribution $p_g(\mathbf{x})$ will converge to the desired target distribution $p^+(\mathbf{x})$.

Theorem 1. *When $D(\cdot)$ and $G(\cdot)$ are updated according to* (6) *and* (7)*, the optimal discriminator is*

$$D^*(\mathbf{x}) = \frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x}) + \lambda p^-(\mathbf{x})}; \tag{8}$$

Moreover, by supposing the support of target and contamination distributions are disjoint, i.e., $Supp(p^+(\mathbf{x})) \cap Supp(p^-(\mathbf{x})) = \emptyset$ and $\lambda \to +\infty$, the generator distribution $p_g(\mathbf{x})$ will converge to the target distribution $p^+(\mathbf{x})$.

Proof. Please refer to the Supplementary Materials. □

Here, we provide a sketch of the proof to gain some insights to the theorem. By deriving the derivatives of $V(D)$ in (6) w.r.t. $D(\cdot)$ and setting it to 0, the optimal discriminator $D^*(\mathbf{x})$ in (8) can be easily obtained. To see the convergence result $p_g(\mathbf{x}) \to p^+(\mathbf{x})$, substituting $p_d(\mathbf{x}) = \pi p^+(\mathbf{x}) + (1-\pi)p^-(\mathbf{x})$ and the optimal discriminator $D^*(\mathbf{x})$ in (8) into the $V(G)$ in (7) gives

$$\begin{aligned} &V(G) \\ &= \int_{\mathcal{S}}\Bigg\{\pi\left(\frac{\pi p^+ + (1-\pi)p^-}{\pi p^+ + (1-\pi)p^- + \lambda p^- + p_g} - c\right)^2 p^+ \\ &+ (1-\pi)\left(\frac{\pi p^+ + (1-\pi)p^-}{\pi p^+ + (1-\pi)p^- + \lambda p^- + p_g} - c\right)^2 p^- \\ &+ \left(\frac{\pi p^+ + (1-\pi)p^-}{\pi p^+ + (1-\pi)p^- + \lambda p^- + p_g} - c\right)^2 p_g \\ &+ \left(\frac{\pi p^+ + (1-\pi)p^-}{\pi p^+ + (1-\pi)p^- + \lambda p^- + p_g} - c\right)^2 p^-\Bigg\}\mathrm{d}\boldsymbol{x}, \end{aligned} \tag{9}$$

where the argument in the distribution is omitted for conciseness, *e.g.*, $p^+(\mathbf{x})$ is abbreviated as p^+; and $\mathcal{S}$ represents the entire real space. From the assumption that the supports of $p^+(\mathbf{x})$ and $p^-(\mathbf{x})$ are disjoint, we have $p^-(\mathbf{x}) = 0$ when $p^+(\mathbf{x}) > 0$. Similarly, we must have $p^+(\mathbf{x}) = 0$ when $p^-(\mathbf{x}) > 0$. We thus divide the whole space $\mathcal{S}$ into two non-overlapped sub-spaces $\mathcal{S}_1$ and $\mathcal{S}_2$, with $p^+(\mathbf{x}) \geq 0$ and $p^-(\mathbf{x}) = 0$ in $\mathcal{S}_1$, and $p^-(\mathbf{x}) \geq 0$ and $p^+(\mathbf{x}) = 0$ in $\mathcal{S}_2$. Based on this observation, (9) can be simplified as

$$\begin{aligned} V(G) = &\int_{\mathcal{S}_1}\left\{\left(\frac{\pi p^+}{\pi p^+ + p_g} - c\right)^2(\pi p^+ + p_g)\right\}\mathrm{d}\boldsymbol{x} \\ &+ \int_{\mathcal{S}_2}\Bigg\{(2-\pi)\left(\frac{(1-\pi)p^-}{(1-\pi)p^- + \lambda p^- + p_g} - c\right)^2 p^- \\ &+ \left(\frac{(1-\pi)p^-}{(1-\pi)p^- + \lambda p^- + p_g} - c\right)^2 p_g\Bigg\}\mathrm{d}\boldsymbol{x}. \end{aligned} \tag{10}$$

When λ is set to be a very lage number, we can see that $\frac{(1-\pi)p^-(\mathbf{x})}{(1-\pi)p^-(\mathbf{x}) + \lambda p^-(\mathbf{x}) + p_g(\mathbf{x})}$ converges to 0. Based on this observation, (10) can be further written as

$$\begin{aligned} V(G) = &\int_{\mathcal{S}_1}\left\{\left(\frac{\pi p^+}{\pi p^+ + p_g} - c\right)^2(\pi p^+ + p_g)\right\}\mathrm{d}\boldsymbol{x} \\ &+ \int_{\mathcal{S}_2}\left\{c^2(2-\pi)p^- + c^2 p_g\right\}\mathrm{d}\boldsymbol{x}. \end{aligned} \tag{11}$$

Define the function $\varphi(x) \triangleq (x-c)^2$. Obviously, $\varphi(x)$ is a convex function, thus according to Jensen inequality, we must have $\varphi\Big(\int_{-\infty}^{\infty} g(x)p(x)\mathrm{d}x\Big) \leq \int_{-\infty}^{\infty} \varphi\left(g(x)\right)p(x)\ \mathrm{d}x$ for any distribution $p(x)$ and function $g(\cdot)$. By denoting $\int_{\mathcal{S}_1} p_g \mathrm{d}\boldsymbol{x}$ as α, then we must have $\int_{\mathcal{S}_2} p_g \mathrm{d}\boldsymbol{x} = 1-\alpha$. Combining with the inequality above, we can infer from (11) that

$$V(G) \geq (\pi + \alpha)\,\varphi\left(\frac{\pi}{\pi + \alpha}\right) + c^2(3 - \pi - \alpha). \tag{12}$$

It can be easily shown that the r.h.s. of (12) is a monotonic decreasing function of α, thus the r.h.s. of (12) is minimized when α is equal to 1. Thus, the inequality

$$V(G) \geq (1+\pi)\,\varphi\left(\frac{\pi}{1+\pi}\right) + c^2\,(2-\pi) \tag{13}$$

always holds, which is obtained by setting $\alpha = 1$. On the other hand, if we substitute $p_g = p^+$ into (11), we obtain

$$V(G) = (1+\pi)\,\varphi\left(\frac{\pi}{1+\pi}\right) + c^2\,(2-\pi)\,. \tag{14}$$

Comparing (14) to (13), we can see that $V(G)$ attains its global minima when $p_g = p^+$. Therefore, if $D(\cdot)$ and $G(\cdot)$ are updated according to (6) and (7), under the specified conditions, the generator distribution will converge to the target distribution $p^+(\mathbf{x})$. The rigorous and detailed proof is given in the Supplementary.

In practice, the disjoint support condition $Supp(p^+(\mathbf{x})) \cap Supp(p^-(\mathbf{x})) = \emptyset$ could be considered being satisfied when the desired and contamination instances look sufficiently different. But when they share many similarities, the generator's ability of only generating desired instances could be compromised. For the parameter λ, theoretically, it should be set very large. But in practice, since we may not be able to find a classifier to separate the target and contamination instances, if λ is set too large, the classifier is likely to classify all instances to 0, which, obviously, will weaken the discriminator's ability of distinguishing between the target and contamination instances. Hence, there should be a balance on the choice of λ. We observe that it only has a minor influence on the performance as long as it is not set too large or too small (*e.g.,* $1 \leq \lambda \leq 5$). We simply set it to 1 in all experiments of this paper.

PuriGAN with Three-Level Discriminator

For the discriminator of two-level PuriGAN, it outputs the same value 0 for both contamination instances from $p^{-}(\mathbf{x})$ and generated instances from $p_g(\mathbf{x})$, making it lack the ability to distinguish between the two types of insances. To further improve the generation performance, we propose to further augment the discriminator by requiring it output three different values for instances from $\mathcal{X}$, $\mathcal{X}^{-}$ and $p_g(\mathbf{x})$, respectively. To this end, we propose to update the discriminator as follows

$$\begin{aligned}\min_D V(D) =& \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\left[(D(\mathbf{x})-1)^2\right] \\ &+ \mathbb{E}_{\mathbf{x}\sim p_g(\mathbf{x})}\left[(D(\mathbf{x})-0)^2\right] \\ &+ \mathbb{E}_{\mathbf{x}\sim p^{-}(\mathbf{x})}\left[(D(\mathbf{x})-d)^2\right], \end{aligned} \tag{15}$$

where d is a value that will be specifically set later. Since the discriminator is designed to output three different values, it should possess some ability to distinguish the three types of instances. With the augmented discriminator derived from (15), the generator can be trained by encouraging the discriminator $D(\cdot)$ to output c for all instances, that is,

$$\begin{aligned}\min_G V(G) =& \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\left[(D(\mathbf{x})-c)^2\right] \\ &+ \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[(D(G(\mathbf{z}))-c)^2\right] \\ &+ \mathbb{E}_{\mathbf{x}\sim p^{-}(\mathbf{x})}\left[(D(\mathbf{x})-c)^2\right], \end{aligned} \tag{16}$$

where c could be any value within $(0, 1)$. We can also prove that under some mild conditions, the generator distribution $p_g(\mathbf{x})$ will converge to the target distribution.

Theorem 2. *When $D(\cdot)$ and $G(\cdot)$ are updated according to* (15) *and* (16)*, the optimal discriminator is*

$$D^{*}(\mathbf{x}) = \frac{p_d(\mathbf{x}) + d \cdot p^{-}(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x}) + p^{-}(\mathbf{x})}; \tag{17}$$

Moreover, if d is set as

$$d = \frac{2\pi - 1}{\pi + 1}, \tag{18}$$

the generator distribution $p_g(\mathbf{x})$ will converge to the target distribution $p^{+}(\mathbf{x})$.

Proof. Please refer to the Supplementary Materials. □

The proof of Theorem 2 is very different from that of Theorem 1 since the subtle premises of disjoint support $Supp(p^{+}(\mathbf{x})) \cap Supp(p^{-}(\mathbf{x})) = \emptyset$ and infinite weighting parameter λ in Theorem 1 are not required. The key to prove Theorem 2 is to find an appropriate value for d such that the generator distribution $p_g(\mathbf{x})$ induced from the updating equations equals to target distribution $p^{+}(\mathbf{x})$. We refer readers to the Supplementary for detailed and rigorous proof.

Theorem 2 does not rely on the subtle disjoint support and infinite weighting parameter λ condition, but it requires to know the ratio of desired instances π. In practice, despite it is difficult to know the exact value of π, many methods have been developed to estimate the ratio that a type of instances accounts for in a dataset (Ramaswamy, Scott, and Tewari 2016; Christoffel, Niu, and Sugiyama 2016; Jain, White, and Radivojac 2016). Thus, we can use these existing methods to estimate the value of π. To evaluate the influence of using a estimated π, we conduct sensitive analysis to the parameter π in our experiments, revealing that the generation performance is not sensitive to small estimation error of π. Thus, an estimate of π is enough to deliver a competitive performance for PuriGAN. It is also worth pointing out that under the special case of $\pi = 0.5$, we can see that the updating rules of $G(\cdot)$ and $D(\cdot)$ in three-level PuriGAN are exactly the same as those in two-level PuriGAN with $\lambda = 1$.

Applications

In addition to generate novel instances from $p^{+}(\mathbf{x})$, PuriGAN can be applied to lots of downstream tasks. In this section, two examples of them are demonstrated.

Anomaly Detection on Contaminated Datasets

One of the widely used anomaly detection approaches are to train a generative model (GAN (Goodfellow et al. 2014) or VAE (Kingma and Welling 2014)) on a dataset that only contains normal samples to capture the distribution of normal samples. Anomalies can then be detected using the criteria derived from the generative models, such as the density value, reconstruction error (Schlegl et al. 2017; Zenati et al. 2018; Akcay, Atapour-Abarghouei, and Breckon 2018) etc. However, in many scenarios, the collected dataset is often mixed with a proportion of anomalous instances. Thus, if we directly train a generative model on the contaminated dataset, it would learn the mixture distribution, compromising its ability to detect anomalies. A remedy to this is to collect a small number of anomalies and then train PuriGAN to only learn the distribution of normal instances. In our experiments, we employ the output value of PuriGAN's discriminator as the detection criteria by noticing that the value partially indicates the normality of input instance. Obviously, more sophisticated criteria can be used, such as leveraging the reconstruction error etc. But this is beyond the focus of this paper, hence we leave it for future work.

PU-learning

The task of PU-learning is to classify an unlabelled dataset into two classes when an extra dataset containing only one class of instances is available (Kiryo et al. 2017). Obviously,

Data	LSGAN	NDA	GenPU	Rumi-LSGAN	PU-LSGAN	PU-NDA	PuriGAN$_1$	PuriGAN$_2$
MNIST	26.93 ± 5.7	21.94 ± 8.9	21.28 ± 6.0	13.31 ± 2.4	15.21 ± 3.2	10.35 ± 3.1	9.71 ± 1.9	**9.51 ± 1.0**
F-MNIST	62.44 ± 6.8	58.88 ± 14.3	58.73 ± 8.0	37.24 ± 4.4	46.94 ± 5.1	44.43 ± 5.4	37.61 ± 4.6	**34.86 ± 3.2**
SVHN	28.50 ± 4.9	27.58 ± 5.2	26.87 ± 5.6	21.08 ± 2.5	26.44 ± 3.9	23.21 ± 3.5	20.32 ± 2.4	**19.63 ± 3.6**
CelebA	49.29 ± 1.9	52.93 ± 2.3	45.81 ± 1.8	42.37 ± 1.9	44.75 ± 2.3	46.34 ± 2.8	36.43 ± 1.5	**35.67 ± 1.5**
CIFAR-10	61.08 ± 9.9	72.95 ± 10.7	62.59 ± 10.8	56.28 ± 10.7	59.26 ± 10.7	70.12 ± 11.8	54.70 ± 10.4	**52.70 ± 10.6**

Table 1: Comparison of FID scores↓ on different datasets under $\gamma_p = 0.4$ and $\gamma_c = 0.2$, where PuriGAN$_1$ and PuriGAN$_2$ denote PuriGAN using two- and three-level discriminator, respectively.

the setting of PU-learning fits with PuriGAN naturally. To perform PU-learning, we propose to train the two-level PuriGAN on the two provided datasets and then use the discriminator to classify the unlabelled instances by noticing that the discriminator is designed to distinguish between the two types of instances. Comparing to traditional PU-learning methods that directly train a classifier, the generation part in PuriGAN plays a role of data augmentation by generating new training samples, which is potentially able to lead to a more competitive performance.

Related Work

Generative adversarial networks are known for their strong capability to generate realistic-looking samples through adversarial training (Goodfellow et al. 2014). Since the seminal work of GAN, many kinds of variants of GAN have been proposed to further improve its modeling ability and training stability by using new model architectures (Radford, Metz, and Chintala 2016; Karras et al. 2018; Brock, Donahue, and Simonyan 2018; Karras, Laine, and Aila 2019), different distance metrics or divergences (Arjovsky, Chintala, and Bottou 2017; Nowozin, Cseke, and Tomioka 2016; Mao et al. 2017) and novel training techniques (Miyato et al. 2018; Wu et al. 2021). Although these models improve the generation performance or training stability of GANs significantly, they are all established on the assumption that all instances in the training datasets are desirable. When facing with contaminated datasets, they do not have the ability to counter the influences of contamination instances. Two recent works that are relevant to our PuriGAN are the negative data augmentation (NDA) (Sinha et al. 2020) and Rumi-GAN (Asokan and Seelamantula 2020), which both explicitly teach a GAN what not to learn by leveraging an extra negative dataset that are composed of undesired instances. Although the two models also make use of an extra negative dataset during the training, their primary goal of using negative dataset is to make GANs avoid generating undesired instances due to the strong generalization abilities of GANs. However, they are still established on the assumption that the training dataset is clean and thus cannot deal with the scenarios with contaminated datasets.

Another line of works that are relevant to our paper is PU-learning (Kiryo et al. 2017; Du Plessis, Niu, and Sugiyama 2015; Kato, Teshima, and Honda 2018), which aim to classify an unlabelled dataset by leveraging an extra dataset solely composed of one class of instances. Recently, some works have proposed to leverage the generation ability of GANs to perform this task. For examples, GenPU in (Hou et al. 2018) proposed to train an array of generators and discriminators to distinguish between the positive and negative instances in the unlabelled datasets. PAN (Hu et al. 2021) proposed to train the PU-learning classifier under the GAN framework by viewing instances selected by the classifier as the generated instances. However, these works generally focus on how to obtain a better classifier/discriminator rather than how to generate high-quality desired instances.

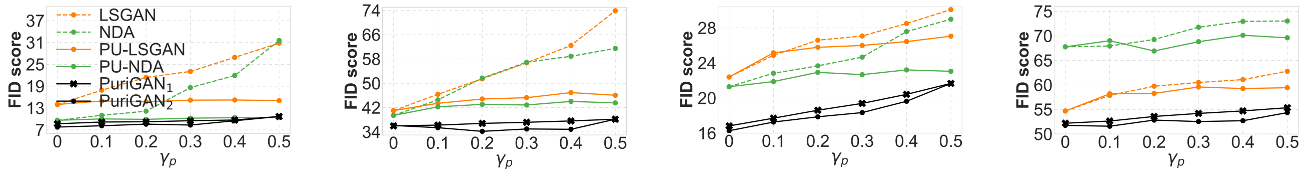
Experiments

Experimental Setups

Evaluation To evaluate the generation performance of PuriGAN[1], for datasets MNIST, F-MNIST, SVHN and CIFAR-10, we randomly select one category and view its instances as the desired instances, while viewing a proportion of instances from another five categories randomly selected from the remaining nine as contamination instances. The two types of instances constitute the final contaminated datasets for training. For dataset CelebA, since it does not have a label, it is partitioned into two subsets according to its attribute value 'bald', with images from each subset viewed as desired and contaminated instances, respectively. For each contaminated dataset, the number of desired instances is fixed, while the number of contamination instances is controlled by the contamination ratio γ_p, which is defined as the ratio between the number of contamination instances and total instances in the training dataset. On the other hand, the number of available contamination instances is controlled by parameter γ_c, which is defined as the ratio between the number of available contamination instances and total instances in the training dataset. The trained models are evaluated on the desired instances in testing dataset with the widely used criteria of Fréchet inception distance (FID) (Heusel et al. 2017), which is computed by following the protocol in (Asokan and Seelamantula 2020). For each dataset, we repeat the random selections and training processes for ten times and the averaged results are reported as the final performance.

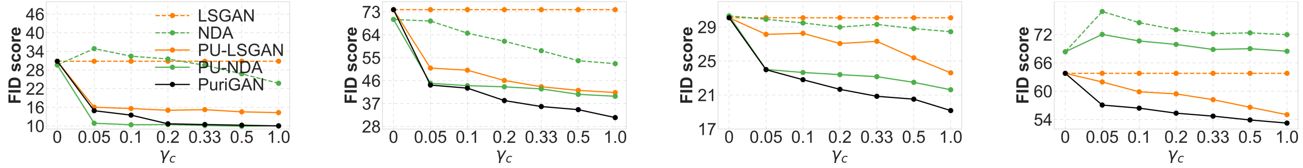
Baselines We compare PuriGAN with the following baselines. 1) *LSGAN* (Mao et al. 2017): it is developed for clean datasets and is not able to leverage the collected contamination instances; 2) *NDA* (Sinha et al. 2020): it is able to leverage the collected contamination instances to boost generation quality, but it is also developed to only work on clean datasets; 3) *GenPU* (Hou et al. 2018): it is a GAN-based PU-learning method that is partially able to work on contami-

[1] Pytorch code is available at https://github.com/tbw162/PuriGAN and Mindspore code is available at https://github.com/tbw162/PuriGAN-mindspore.



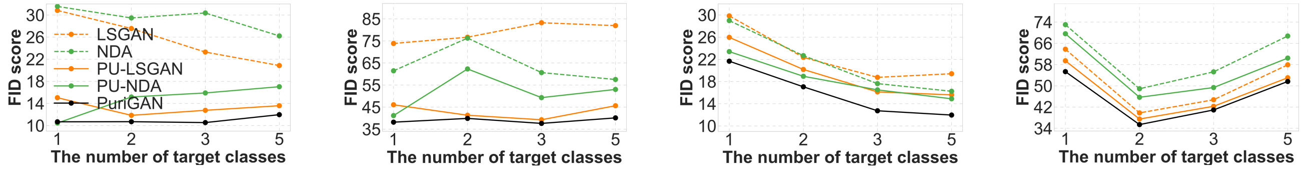
MNIST F-MNIST SVHN CIFAR-10

Figure 1: FID score as a function of contamination ratio γ_p under a fixed $\gamma_c = 0.2$, where PuriGAN$_1$ and PuriGAN$_2$ denotes PuriGAN using two-level and three-level discriminator, respectively.



MNIST F-MNIST SVHN CIFAR-10

Figure 2: FID score as a function of γ_c, the ratio between the collected contamination instances and target instances, under a fixed $\gamma_p = 0.5$.



MNIST F-MNIST SVHN CIFAR-10

Figure 3: FID score as a function of the number of classes of target instances under the scenario of $\gamma_p = 0.5$ and $\gamma_c = 0.2$.

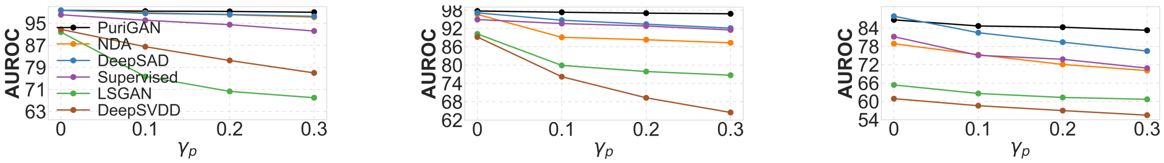
nated datasets. 4) *Rumi-LSGAN* (Asokan and Seelamantula 2020): It can also leverage the contamination instances but requires the dataset $\mathcal{X}$ to be clean. 5) *PU-LSGAN*: it is a two-stage training method by combining PU-learning and LSGAN, in which the PU-learning employs the recently proposed nnPU method (Kiryo et al. 2017); 6) *PU-NDA*: it is a combination of PU-learning and NDA;

Performance on Image Generation

Table 1 presents the FID scores of the proposed PuriGANs and comparable models on four datasets under the specific contamination ratio $\gamma_p = 0.4$ and available contamination ratio $\gamma_c = 0.2$. From the table, it can be observed that the proposed PuriGANs perform significantly better than LSGAN and NDA, which demonstrates that the proposed PuriGANs are effective to counter the influence of contamination instances in the training datasets comparing to traditional GANs. Moreover, comparing to the two-stage methods that employ PU-learning to remove contamination instances first, we can see that these two-stage methods still lag behind our proposed PuriGANs, especially on relatively complex datasets F-MNIST and CIFAR-10. This is because PU-learning tends to perform well on simple datasets, but unsatisfactorily on complex ones, resulting in relatively small improvements on complex datasets.

Impact of Contamination Ratio γ_p Fig. 1 shows how FID scores vary as a function of the contamination ratio γ_p when the available contamination ratio γ_c is fixed to 0.2. It can be seen that as γ_p increases, the FID scores for LSGAN and NDA steadily increases on all four datasets, too, which implies the decrease of generated images' quality. This is easy to understand because the two GANs do not have any ability to counter the influences of contamination instances. By incorporating the PU-learning, we can see that the FID scores of both PU-LSGAN and PU-NDA increase much slowly, which indicates the effectiveness of the two-stage training methods in countering the influence of contamination. Comparing the two-stage methods to our PuriGANs, we can see that our models still yield better FID scores, which may be partially attributed to the joint consideration of purification and generation in our proposed PuriGAN. Lastly, we can see that PuriGAN$_2$ overall performs better than PuriGAN$_1$. This is in consistent with our expectation because PuriGAN with three-level discriminator does not rely on the subtle disjoint support condition. In addition, we also see that the two PuriGANs have the same FID score at $\gamma_p = 0.5$. That is because the updating rules of the two models become the same at this special case.

Impact of Available Contamination Ratio γ_c Fig. 2 shows how FID scores vary as a function of γ_c when the contamination ratio γ_p is fixed to 0.5. Due to $\gamma_p = 0.5$, the two PuriGANs become equivalent, thus we only illustrate the performance of one PuriGAN. From Fig. 2, it can be seen that the performance of PuriGAN can be steadily improved as more contamination instances are collected on all considered datasets, demonstrating the effectiveness of PuriGAN in leveraging the extra contamination instances to counter the influences of contamination. Although the two-



MNIST

F-MNIST

CIFAR-10

Figure 4: AUROC of semi-supervised anomaly detection under different contamination ratio of γ_p and fixed $\gamma_c = 0.05$.

stage methods PU-LSGAN and PU-NDA can also benefit from more available contamination instances, they are not as effective as the proposed PuriGAN on relatively complex datasets F-MNIST and CIFAR-10.

Impact of the Number of Classes of Target Instances In the previous experiments, we only use instances from one class as the target instances. In this section, we investigate how the generation performance is affected when the target instances are composed of more classes of instances. Fig. 3 shows how the FID scores vary as the number of classes of target instances under the scenario of $\gamma_p = 0.5$ and $\gamma_c = 0.2$. It can be seen from Fig. 3 that the proposed PuriGAN still performs the best among all comparable baselines under the scenario with more classes of target instances.

Performance on Downstream Tasks

Anomaly Detection with Contaminated Datasets For evaluation, we select instances from one category and a proportion of instances from the remaining categories to construct a contaminated dataset $\mathcal{X}$. Moreover, we also assume the availability of another dataset $\mathcal{X}^-$ that is only composed of instances from the remaining categories. The goal of this task is to detect the anomalies, which are those belonging to the remaining categories, by leveraging a model trained on $\mathcal{X}$ and $\mathcal{X}^-$. As discussed in Applications section, the output value of PuriGAN's discriminator can be used to detect anomalies. To better evaluate PuriGAN, we compare it with several representative unsupervised methods, *Deep SVDD* (Ruff et al. 2018), *LSGAN* (Mao et al. 2017), and semi-supervised methods, the recently developed *Deep SAD* (Ruff et al. 2019) and *NDA* (Sinha et al. 2020), which can leverage the collected dataset $\mathcal{X}^-$ for detection. In addition, we also train a binary classifier by treating instances from $\mathcal{X}$ and $\mathcal{X}^-$ as 1 and 0, respectively, and then use it to detect anomalies. The area under receiver operating characteristic curve (AUROC) is used as the evaluation criteria. For each setting, the experiments were run ten times, and their average is reported as the final performance. From Figure 4, it can be observed that the PuriGAN-based method has a very stable performance across different ratios of contamination in the training dataset. By contrast, the performances of all compared unsupervised and semi-supervised methods deteriorate steadily as γ_p increases. This further corroborates the advantages of our proposed PuriGAN in countering the influences of contamination in the training datasets.

PU-learning Following the setups in the paper PAN (Hu et al. 2021), the proposed PuriGAN is evaluated on two text and two image datasets on this task. F1-score and accuracy are employed as the performance criteria. From Table 2, it can be seen that PuriGAN outperforms all baselines on the considered datasets. This is probably because the generator in PuriGAN approximately plays a role of data augmentation by generating new training samples continuously, thereby leading to a more competitive performance comparing with traditional PU-learning methods that directly train a classifier.

Dataset	20News		IMDB		MNIST		CIFAR-10	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
a-GAN	63.5	68.7	73.0	70.6	94.7	95.0	76.2	83.1
UPU	59.1	53.0	70.4	69.9	94.2	94.3	86.2	89.0
nnPU	78.5	78.1	76.2	74.6	95.4	95.4	86.1	88.4
nnPUSB	75.9	75.6	74.2	71.9	95.6	95.6	86.6	88.6
PAN	81.1	81.1	77.1	78.8	96.5	96.4	87.2	89.7
PuriGAN	**85.7**	**84.6**	**79.7**	**78.9**	**96.8**	**96.9**	**88.7**	**90.9**

Table 2: F1 score and accuracy of different PU-learning methods.

Conclusion

In this paper, we studied the problem of how to train GANs to only generate target instances when a contaminated training dataset is presented. To this end, with the introduction of another extra dataset composed of only contamination instances, a purified generative adversarial network framework (PuriGAN) is proposed, which is achieved by augmenting the discriminator in traditional GANs to endow it with the ability to distinguish between the desired and undesired instances. We prove that the proposed PuriGANs are guaranteed to converge to the target distribution under some mild conditions. Extensive experiments are conducted to demonstrate the superior performance of the proposed PuriGANs in generating images only from desired categories. Moreover, we also apply it to the downstream tasks of semi-supervised anomaly detection and PU-learning, which shows that PuriGAN can deliver the best performance over comparable baselines on both tasks.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62276280, U1811264, 62276279), Key R&D Program of Guangdong Province (No. 2018B010107005), Natural Science Foundation of Guangdong Province (No. 2021A1515012299), Science and Technology Program of Guangzhou (No. 202102021205), and CAAI-Huawei MindSpore Open Fund.

References

Akcay, S.; Atapour-Abarghouei, A.; and Breckon, T. P. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, 622–637.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223.

Asokan, S.; and Seelamantula, C. 2020. Teaching a gan what not to learn. *Advances in Neural Information Processing Systems*, 33: 3964–3975.

Bekker, J.; and Davis, J. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4): 719–760.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*.

Christoffel, M.; Niu, G.; and Sugiyama, M. 2016. Class-prior estimation for learning from positive and unlabeled data. In *Asian Conference on Machine Learning*, 221–236.

Du Plessis, M.; Niu, G.; and Sugiyama, M. 2015. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*, 1386–1394.

Elkan, C.; and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 213–220.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Hou, M.; Chaib-Draa, B.; Li, C.; and Zhao, Q. 2018. Generative adversarial positive-unlabeled learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2255–2261.

Hu, W.; Le, R.; Liu, B.; Ji, F.; Ma, J.; Zhao, D.; and Yan, R. 2021. Predictive adversarial learning from positive and unlabeled data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7806–7814.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Izadi, S.; Mirikharaji, Z.; Kawahara, J.; and Hamarneh, G. 2018. Generative adversarial networks to segment skin lesions. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 881–884. IEEE.

Jain, S.; White, M.; and Radivojac, P. 2016. Estimating the class prior and posterior from noisy positives and unlabeled data. *Advances in neural information processing systems*, 29.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kato, M.; Teshima, T.; and Honda, J. 2018. Learning from positive and unlabeled data with a selection bias. In *International conference on learning representations*.

Kazeminia, S.; Baur, C.; Kuijper, A.; van Ginneken, B.; Navab, N.; Albarqouni, S.; and Mukhopadhyay, A. 2020. GANs for medical image analysis. *Artificial Intelligence in Medicine*, 109: 101938.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning*.

Kiryo, R.; Niu, G.; Du Plessis, M. C.; and Sugiyama, M. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30.

Liu, H.; Wan, Z.; Huang, W.; Song, Y.; Han, X.; and Liao, J. 2021. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9371–9381.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Paul Smolley, S. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2794–2802.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*.

Northcutt, C.; Jiang, L.; and Chuang, I. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70: 1373–1411.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations*.

Ramaswamy, H.; Scott, C.; and Tewari, A. 2016. Mixture proportion estimation via kernel embeddings of distributions. In *International conference on machine learning*, 2052–2060.

Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S. A.; Binder, A.; Müller, E.; and Kloft, M. 2018. Deep one-class classification. In *International conference on machine learning*, 4393–4402.

Ruff, L.; Vandermeulen, R. A.; Görnitz, N.; Binder, A.; Müller, E.; Müller, K.-R.; and Kloft, M. 2019. Deep Semi-Supervised Anomaly Detection. In *International Conference on Learning Representations*.

Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; and Langs, G. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *IPMI*, 146–157.

Sinha, A.; Ayush, K.; Song, J.; Uzkent, B.; Jin, H.; and Ermon, S. 2020. Negative Data Augmentation. In *International Conference on Learning Representations*.

Wu, Y.-L.; Shuai, H.-H.; Tam, Z.-R.; and Chiu, H.-Y. 2021. Gradient normalization for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6373–6382.

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5505–5514.

Zenati, H.; Romain, M.; Foo, C.-S.; Lecouat, B.; and Chandrasekhar, V. 2018. Adversarially learned anomaly detection. In *2018 IEEE International conference on data mining (ICDM)*, 727–736.