IMPROVING ANOMALY DETECTION WITH A SELF-SUPERVISED TASK BASED ON GENERATIVE ADVERSARIAL NETWORK

Heyan Chai [1], *Weijun Su*[1], *Siyu Tang*[1], *Ye Ding*[2], *Binxing Fang*[1], *Qing Liao*[1,3]✉

[1] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
[2] School of Cyberspace Security, Dongguan University of Technology
[3] Peng Cheng Laboratory, Shenzhen, China,

ABSTRACT

Existing anomaly detection models show success in detecting abnormal images with generative adversarial networks on the insufficient annotation of anomalous samples. However, existing models cannot accurately identify the anomaly samples which are close to the normal samples. We assume that the main reason is that these methods ignore the diversity of patterns in normal samples. To alleviate the above issue, this paper proposes a novel anomaly detection framework based on generative adversarial network, called ADe-GAN. More concretely, we construct a self-supervised learning task to fully explore the pattern information and latent representations of input images. In model inferring stage, we design a new abnormality score approach by jointly considering the pattern information and reconstruction errors to improve the performance of anomaly detection. Extensive experiments show that the ADe-GAN outperforms the state-of-the-art methods over several real-world datasets.

Index Terms— Anomaly detection, self-supervision, interpolation, generative adversarial networks

1. INTRODUCTION

Anomaly detection aims to recognize whether a novel sample is an inlier or an outlier [1]. Improving the capability of anomaly detection is an important problem and receives significant attentions in many real-world application areas such as medical diagnosis [2, 3], drug discovery [4], cybersecurity [5], and computer vision applications [6, 7].

Existing methods achieves promising performance in anomaly detection of images, with the assumption that abnormal samples bring larger reconstruction errors than normal samples when reconstructs sample images [3, 8, 1, 9, 10, 11]. Some methods primarily leverage generative adversarial networks [12] or deep encoder-decoder networks [13] to compute a reconstruction error which is utilized to calculate the abnormality score for detecting the anomaly. For example, AnoGAN [3] is first to use generative adversarial network (GAN) to reconstruct the images and then to detect the anomaly through the reconstruction errors. Introducing Bi-GAN to train an encoder which reduces the higher computational cost in AnoGAN architecture [8, 14]. GPND [1] uses a probabilistic approach to effectively compute the reconstruction errors between normal and abnormal samples by employing an autoencoder architecture.

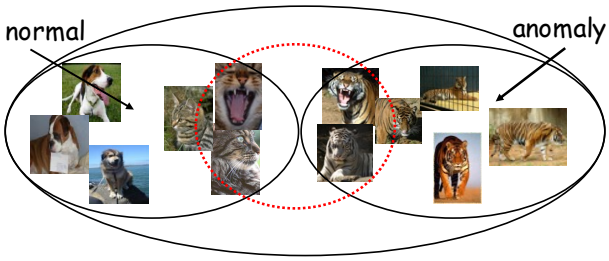


Fig. 1. Illustration of anomaly samples close to normal ones.

Although the above approaches have their fair share of success, most existing methods treat the anomaly detection task as a binary classification problem that tries to classify samples as normal and abnormal only by reconstruction errors. However, ignoring different patterns of normal data and treating all normal samples as one class may decrease detection performance in real-world applications. For example, as shown in **Fig. 1**, we regard *"dog"* and *"cat"* images as normal samples and *"tiger"* images as abnormal samples. It can be observed that there are many types of patterns in normal samples such as *"dog"* and *"cat"*, and the differences between patterns are very large. Moreover, we can find that existing anomaly detection methods are difficult to distinguish normal sample *"cat"* and abnormal sample *"tiger"* in the red dashed circle based only on reconstruction errors, because the sample distribution of *"cat"* is more similar[1] to *"tiger"* rather than *"dog"*. In summary, similar distributions between some abnormal samples and normal samples can lead to a smaller reconstruction error, which significantly limits the performance of existing detection methods.

Corresponding author: Qing Liao (Email:liaoqing@hit.edu.cn).

[1]Two distributions are similar when they have the similar mean value, or even has a smaller mean square error.

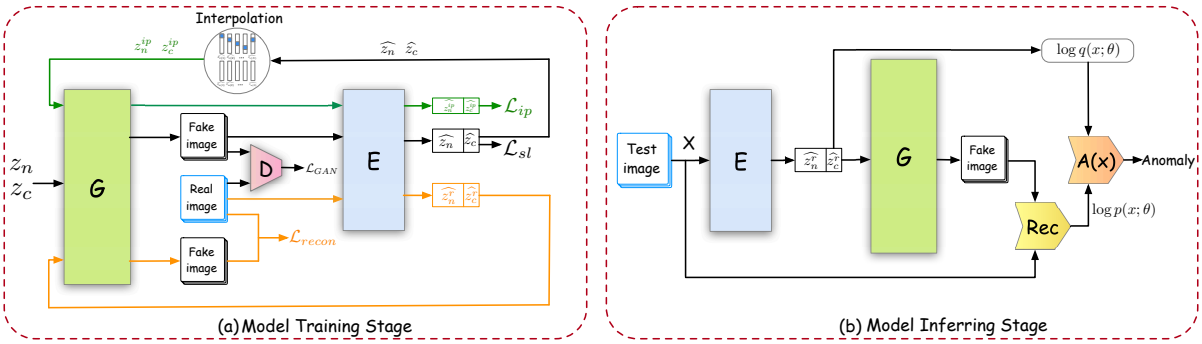


Fig. 2. (a) Training procedure of ADe-GAN and (b) anomaly detection by using proposed abnormality score mechanism.

To address above issues, we argue that a more effective way to improve anomaly detection performance depends on considering the diversity of patterns in normal samples rather than treating the patterns in normal data as one pattern. Inspired by GAN [15], we propose a novel **A**nomaly **De**tection **G**enerative **A**dversarial **N**etwork framework (**ADe-GAN**) to improve anomaly detection via exploring the pattern information in normal data. More concretely, we first construct a new pattern-related self-supervised learning task in ADe-GAN to learn the pattern information of input images via an extractor. Then, for better distinguishing the abnormal samples, we design a novel abnormality score mechanism to identify the different abnormal samples by jointly considering reconstruction error and pattern information. Specifically, we use reconstruction error to identify the abnormal samples far away from normal samples, and leverage pattern information of samples to identify the abnormal samples close to the normal samples.

2. PROBLEM FORMULATION

We give a formal description about anomaly detection problem to help understand this problem.

Definition 2.1 (Anomaly Detection) *Given a data set $\mathcal{D}$ consists of N normal images, $\mathcal{D} = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, which are sampled from normal data distributions $P_n(x)$, where $x \in \mathbb{R}^n$, and a testing data set $\widehat{\mathcal{D}}$ with M images, $\widehat{\mathcal{D}} = \{(\widehat{x}^{(1)}, y_1), (\widehat{x}^{(2)}, y_2), \ldots, (\widehat{x}^{(M)}, y_M)\}$,where $y_i = 0$ or 1 indicates x_i is sampled from normal data distribution or not. The task is to model $\mathcal{D}$ to learn the manifold of normal image distribution and then detect the samples in $\widehat{\mathcal{D}}$ that are not sample from $P_n(x)$ during the test phase. The goal of the task is to learn a anomaly score function $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^+$.*

3. METHODOLOGY

In this section, we introduce the proposed ADe-GAN and abnormality score mechanism in detail. As depicted in **Fig. 2 (a)**, the architecture of the proposed ADe-GAN contains two components: 1) *generative adversarial network consists of generator G and discriminator D*, which provides a mapping from a latent space to the input space according to the random variables z_c and z_n; 2) *extractor E*, which can learn a proper pattern division scheme to distinguish different images and extract feature representation from input images. **Fig. 2 (b)** presents how to detect anomaly samples via abnormal score mechanism in the model inferring stage.

3.1. Proposed ADe-GAN

3.1.1. *Generative Adversarial Network (GAN)*

We first train the generator G and discriminator D of GAN to make generator fit the normal sample distribution and generate fake images by adversarial learning. After iterative training to convergence, generator can provide a mapping from a latent space to input space and generate the normal sample distribution. That is to say, generator can learn latent pattern information of normal samples and generate different normal image according the different input random variable, z_c, z_n, which is achieved by adversarial learning and pattern-related self-supervised task together (introducing in the next sub-section). We can formally describe the capability of G to fit the normal data distribution by $P_n(x) = G(z_c, z_n)$, where $P_n(x)$ is the normal data distribution. Formally, the adversarial learning objective is defined by:

$$\mathcal{L}_{adv} = \mathop{\mathbb{E}}_{\substack{z_c \in \boldsymbol{e_k} \\ z_n \sim N(0,\sigma^2)}} \log\left(1 - D\left(G\left(z_c, z_n\right)\right)\right) + \mathop{\mathbb{E}}_{X \sim P_n(X)} \log(D(X)) \tag{1}$$

where $z_c \in e_k$ denotes the pattern type of input images and e_k is the k^{th} elementary vector in $\mathbb{R}^K$, $z_n \sim N(0, \sigma^2)$ denotes the feature variable of input images, $z_n \in \mathbb{R}^d$.

3.1.2. *Pattern-Related Self-Supervised Task*

Due to the lack of pattern types of normal data, we construct a pattern-related self-supervised task to obtain the pattern type of normal data by employing an extractor E. One intuitive way to construct a self-supervised task is via the image clustering, with aim to enable extractor output the feature repre-

sentations $\widehat{z}_n$ and pattern types $\widehat{z}_c$ of images. Formally,

$$E(G(z_c, z_n)) = (\widehat{z}_c, \widehat{z}_n) \; or \; E(X) = (\widehat{z}_c, \widehat{z}_n), \quad (2)$$

The extractor can be considered to be a mapping from input space $\mathcal{X}$ to latent space, denoted as $E : \mathcal{X} \mapsto (z_c, z_n)$. The objective function of constructed self-supervised task is defined as:

$$\mathcal{L}_{sl} = \mathop{\mathbb{E}}_{z_c \in \boldsymbol{e_t}, z_n \in N(0,\sigma)} \mathcal{H}(\widehat{z}_c, z_c) + ||\widehat{z}_n - z_n||_2, \quad (3)$$

where $\mathcal{H}$ is cross-entropy loss, $(\widehat{z}_c, \widehat{z}_n) = E(G(z_c, z_n))$, and $\widehat{z}_n$ and $\widehat{z}_c$ are the extracted feature variable and pattern variable of generated samples, respectively.

We introduce interpolation operation [16] to help G better fit the normal data distribution and help E better extract the pattern information of normal data, respectively. The definition of interpolation operation is shown below.

Definition 3.1 (Interpolation Operation) *Suppose samples* $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ *are corresponding to* $\{(z_c^{(1)}, z_n^{(1)}), \ldots, (z_c^{(n)}, z_n^{(n)})\}$ *respectively, the interpolation operation between these samples are defined as* $z_c^{ip} = \sum \lambda_i z_c^{(i)}, z_n^{ip} = \sum \lambda_i z_n^{(i)}$*, where* $\sum \lambda_i = 1$*, and* $\lambda_i \geq 0$*. The corresponding interpolation generated sample are denoted as* X^{ip}*, which can be defined as* $X^{ip} = g(z_c^{ip}, z_n^{ip})$*.*

In order to help G better fit the normal data distribution, we employ *interpolation operation among the same patterns* to achieve the data augmentation (generate samples that did not appear in the training data), which improves the generalization capability of generator. Given subset $\mathcal{D}_s = \{x^{(1)}, \ldots, x^{(t)}\}$ that samples from the same pattern in normal data and its corresponding latent variables $\{(z_c^{(1)}, z_n^{(1)}), \ldots, (z_c^{(t)}, z_n^{(t)})\}$, we can get $z_c^{(1)} = z_c^{(2)} = \cdots = z_c^{(t)}$. Hence, according to the **Definition 3.1**, we can get:

$$z_c^{ip} = \sum \lambda_i z_c^{(i)} = (\sum \lambda_i) z_c^{(1)} = z_c^{(1)}, \quad (4)$$

$$min\{z_n^{(1)}, \ldots, z_n^{(t)}\} \leq z_n^{ip} \begin{array}{l} = \sum \lambda_i z_n^{(i)} \\ \leq max\{z_n^{(1)}, \ldots, z_n^{(t)}\} \end{array}, \quad (5)$$

According to **Eq. (4)** and **(5)**, the interpolated vector $X^{(ip)} = G(z_c^{ip}, z_n^{ip})$ still lies in the same latent space. We can formally describe this process by $P(X^{(ip)}) = P(G(z_c^{ip}, z_n^{ip})) = P(G(z_c, z_n)) = P(X)$. Finally, the generalization capability of G has been further improved.

The G can fit the distribution of normal data very well, but it can not distinguish the different patterns of normal samples. Therefore, we design an extractor E to find and select a effective pattern division scheme to distinguish the normal samples with different patterns and abnormal samples. We employ *interpolation operation among the different patterns* to train E to learn a pattern division scheme that can distinguish the normal and abnormal samples.

Since the distribution of some abnormal samples close to normal samples, E can learn a pattern division scheme that regards the sample after interpolation among different patterns as a normal sample. We expect that interpolation among samples of different patterns can get abnormal sample so that the pattern division scheme can distinguish the normal and abnormal samples. Therefore, we penalize the E by:

$$\mathcal{L}_{ip} = \mathop{\mathbb{E}}_{z_c \notin \boldsymbol{e_t}, z_n} \sum -\frac{1}{c} log \widehat{z}_c, \text{ where } \widehat{z}_c, \widehat{z}_n = E(G(z_c, z_n)), \quad (6)$$

Besides, in order to avoid over-punishing the generator and extractor, we introduce reconstruction loss to stabilize the training procedure, to guarantee $||P(g(z_c, z_n)) - P(X)|| \leq \epsilon$ at the worst situation, which can be defined as:

$$\mathcal{L}_{recon} = \mathbb{E}_{X \sim P_n(X)} ||X - G(E(X))||_2, \quad (7)$$

3.1.3. Learning Objective

The training procedure of ADe-GAN consists of two stages of optimization. The first stage aims to optimize G and D by using the adversarial loss $\mathcal{L}_{adv}$ which is defined by **Eq. (1)**. The target of the second stage is to train G and E by jointly minimizing the three losses defined by **Eq. (3)**, **(6)** and **(7)**:

$$\mathcal{L} = \alpha \mathcal{L}_{ip} + \beta \mathcal{L}_{sl} + \gamma \mathcal{L}_{recon}, \quad (8)$$

where $\alpha, \beta,$ andγ are tuned hyper-parameters.

3.2. Abnormality Score Mechanism

For improving anomaly detection, we design a novel abnormality score mechanism by incorporating reconstruction error and pattern information of samples extracted by E, as shown in **Fig. 2 (b)**. More concretely, we use the normal probability distribution to transform reconstruction error into probability scores to align with patterns probabilities, denoted as:

$$\log p(x; \theta) \approx P_N(||X - G(E(X))||_2) \quad (9)$$

For abnormal samples close to the normal data, the reconstruction error can not distinguish them. Therefore, we use the pattern information to help detect such kind of anomalies. For normal samples, their output probabilities are close to 1, $\text{argmax}(P_{x \in c_i}(x)) = 1$, because extractor can distinguish their pattern types. For abnormal samples, since that they belong to none of the learned patterns, the maximum output probability of patterns is expected to close $\frac{1}{n_c}$, where n_c is the number of patterns. Hence the anomaly probability of pattern information can be defined as :

$$\log q(x; \theta) \approx -\text{argmax}(P_{x \in c_i}(x)), \quad (10)$$

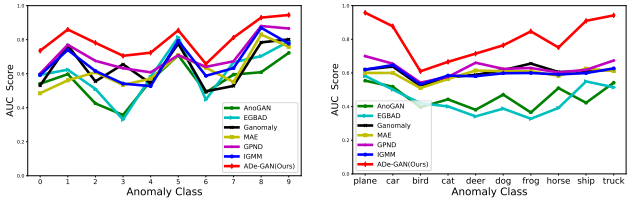
The total anomaly score is defined as:

$$\mathcal{A}(x) = -\frac{1}{n_c} * \log p(x; \theta) + (1 - \log p(x; \theta)) \log q(x; \theta) \quad (11)$$

where n_c denotes the number of patterns.

Table 1. AUC results on MNIST

	AnoGAN	EGBAD	Ganomaly	MAE	GPND	IGMM	**ADe-GAN**
0	0.610	0.755	0.722	0.619	0.943	0.855	**0.971**
1	0.300	0.290	0.468	0.056	0.313	0.408	**0.768**
2	0.535	0.670	0.819	0.662	0.944	0.935	**0.967**
3	0.440	0.520	0.649	0.556	0.873	0.799	**0.923**
4	0.430	0.450	0.677	0.544	**0.908**	0.818	0.888
5	0.420	0.475	0.679	0.600	0.884	0.857	**0.941**
6	0.475	0.570	0.684	0.831	0.868	0.834	**0.928**
7	0.355	0.400	0.571	0.703	0.766	0.655	**0.886**
8	0.400	0.545	0.708	0.798	0.882	0.842	**0.947**
9	0.335	0.345	0.471	0.619	0.601	0.528	**0.852**
average	0.430	0.500	0.645	0.599	0.798	0.756	**0.907**



(a) AUC on Fashion MNIST data set (b) AUC on CIFAR data set

Fig. 3. AUC results of comparison models on two data sets.

4. EXPERIMENTS

4.1. Experimental Setup

Three well-known real-world data sets are used to evaluate the ADe-GAN, including MNIST [17], Fashion MNIST [18] and CIFAR10 [19], and all datasets include ten classes. Following [11, 9, 1], we treat one class of data set as anomaly and the rest of the classes are considered as normal samples. We implement ADe-GAN in PyTorch (v0.4.0 with Python 3.6.5) by optimizing the networks using Adam with learning rate as $2e-4$. The parameters (α, β, γ) are set as $(1, 1, 10)$,$(1, 10, 10)$ for MNIST and CIFAR10 respectively. The parameter n_c is set to 10.

4.2. Experimental Results and Analysis

We compare ADe-GAN with the state-of-the-art methods including AnoGAN [3], EGBAD [8], Ganomaly [11], MAE [9], IGMM [10] and GPND [1]. We use AUC scores to evaluate the performance of anomaly detection.

Performance Evaluation. Table 1 shows AUC scores of seven anomaly detection models on MNIST data set, which selects one from ten categories (from *"0" to "9"*) as the anomaly class and rest are normal samples. The experimental results show that ADe-GAN achieves the best performance compared with other methods and obtains the best average AUC score under all anomaly classes. For example, ADe-GAN outperforms GPND by 10.9%, IGMM by 15.1%, MAE by 30.8% and EGBAD by 40.7% respectively on AUC for MNIST, due to they either ignore the pattern information included in normal data or fail to identify the anomalies that are close to the normal samples. For more complex data sets such as Fashion MNIST and CIFAR, as shown in **Fig. 3**, it is clearly that ADe-GAN significantly outperforms all baselines in all anomaly classes. Specifically, ADe-GAN increases average AUC score by 24.1% compared with AnoGAN and 9.4% by GPND in Fashion MNIST shown in **Fig. 3 (a)**. Besides, **Fig. 3 (b)** shows performance of anomaly detection of all comparison models upon different anomaly classes, which also illustrates the superiority of ADe-GAN.

Fig. 4. Learned patterns when abnormal class is 1.

Visualization of Pattern Information. Fig. 4 presents image visualization results of ten patterns learned by ADe-GAN with setting anomaly class to "1" in MNIST. It is clearly that ADe-GAN can learn pattern information of normal data based on interpolation operation. For each pattern of normal samples shown in each row in **Fig. 4**, each reconstructed handwritten digital image is totally different but the reconstruction error is small, which is caused by interpolation operation among the same pattern. As shown in solid red box in **Fig. 4**, we can clearly recognize the anomaly class due to the huge difference between reconstructed samples caused by the large reconstruction error .

Table 2. AUC score of ADe-GAN under different value of n_c

n_c	2	4	6	8	10	12	14
AUC	0.801	0.856	0.876	0.883	**0.907**	0.899	0.880

Parameter Sensitivity. In this part, we investigate the effect of n_c value in MNIST data set. As shown in **Table 2**, it is clearly that the value of n_c relatively close to 10 facilitates the best AUC score, because the number of classes in MNIST data set is 10. Moreover, too low value of n_c would degrade the performance.

5. CONCLUSION

In this paper, we analyze issues why existing anomaly detection models cannot identify the samples close to normal data. To alleviate above issues, we propose ADe-GAN to improve the anomaly detection performance by constructing a self-supervised task to capture latent pattern information of samples. Besides, by designing a new abnormality score mechanism, ADe-GAN outperforms the state-of-the-art models.

6. ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under grant No. 62076079 and No. U19A2067.

7. REFERENCES

[1] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 6823–6834.

[2] Ruoying Wang, Kexin Nie, Tie Wang, Yang Yang, and Bo Long, "Deep learning for anomaly detection," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 894–896.

[3] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*, 2017, pp. 146–157.

[4] Shaista Hussain, Ayesha Anees, Ankit Das, Binh P. Nguyen, Mardiana Marzuki, Shuping Lin, and et al., "High-content image generation for drug discovery using generative adversarial networks," *Neural Networks*, vol. 132, pp. 353–363, 2020.

[5] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel, "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 190–237, 2014.

[6] Kang Zhou, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, Weixin Luo, Zaiwang Gu, Jiang Liu, and Shenghua Gao, "Encoding structure-texture relation with p-net for anomaly detection in retinal images," in *Proceeding of 16th European Conference on Computer Vision, (ECCV)*, 2020, vol. 12365, pp. 360–377.

[7] Eric Jardim, Lucas A. Thomaz, Eduardo A. B. da Silva, and Sergio L. Netto, "Domain-transformable sparse representation for anomaly detection in moving-camera videos," *IEEE Transactions on Image Processing.*, vol. 29, pp. 1329–1343, 2020.

[8] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar, "Efficient gan-based anomaly detection," *CoRR*, vol. abs/1802.06222, 2018.

[9] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 1705–1714.

[10] Kathryn Gray, Daniel Smolyak, Sarkhan Badirli, and George Mohler, "Coupled igmm-gans for improved generative adversarial anomaly detection," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 2538–2541.

[11] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision (ACCV)*. Springer, 2018, pp. 622–637.

[12] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 38:1–38:38, 2021.

[13] Haoyi Fan, Fengbin Zhang, and Zuoyong Li, "Anomalydae: Dual autoencoder for anomaly detection on attributed networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 5685–5689, IEEE.

[14] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 727–736.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[16] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2016.

[17] Li Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, 2012.

[18] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017.

[19] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," 2009.