A study on Anomaly Detection GAN-based methods on image data

Emanuel H. Silva[1], **Johannes V. Lochter**[2]

[1]Departamento de Engenharia da Computação
Centro Universitário Facens – Sorocaba, SP – Brazil

huber.emanuel@outlook.com, johannes.lochter@facens.br

Abstract. *The anomaly detection task is a well know problem being researched among a variety of areas, including machine learning. The task is to identify data patterns that have a non expected behaviour, that can be a malicious data sent by an attacker or a unexpected valid behaviour, in both cases the anomaly need to be identified. With the advance of deep learning based techniques showing that this class of algorithms can learn high-dimensional and complex data patterns, naturally it became an option to the anomaly detection task. Recent researches in literature are using a sub-field of deep learning algorithms named Generative Adversarial Networks for predicting anomalous samples, since the original method can learn the data distribution. These new techniques make some changes for the anomaly detection task, and this work provides a briefly review on these methods and provides a comparison with well known methods.*

1. Introduction

The data volume generated by users in web and mobile applications is increasing every year at an exponential rate, and based on that data, systems are being developed to learn data patterns and take some action to improve the user experience, make decisions for the business, avoid pitfalls through spam filter, among others. The performance of these systems can be affected if some of the training data used to create it contains some kind of anomaly. The performance of these systems depends a lot on the quality of the data, therefore malicious or corrupted data must be identified, these inconsistencies are also called anomalies.

An anomaly is a pattern or a data point that does not follow the expected behaviour [Chandola et al. 2009]. The behaviour can be explicit model as a set of rules (expert systems) or they can be modeled based on a dataset, for example methods based on clustering [Alguliyev et al. 2017] can be directly used for tabular data, but they struggle when used with image data because the directly use of the distance of the pixels do not yields context information, such as the objects presented in the image.

The detection of anomalies can be applied in a variety of applications and also can play a critical role in a product, e.g. a bank can detect if a user had his credit card stolen or cloned by analyzing his last purchases, if it does not follow the user purchase pattern it can be considered as a potential fraudulent operation and the purchased can be cancelled by the bank, bringing more security to their clients. Another example is an anomalous data traffic pattern in a computer network that could mean that a hacked computer is trying to steal confidential data.

Deep learning algorithms can model complex non-linear data patterns and have been presenting astonishing results in a variety of areas [Brock et al. 2018], [Shetty et al. 2018], [Zhang et al. 2018]. One of deep learning sub-fields is the Generative Adversarial Networks [Goodfellow et al. 2014] technique that learns to model a generator through an adversarial learning, this algorithm have shown great results on capturing the data distribution and reproducing it, such as presented by [Curtó et al. 2017]. Since GANs can learn how to represent the data distribution, some recent researches present methods to use that representation to detect anomalies through a plethora of different approaches. This work aims to expatiate over this recently proposed methods and present some discussion on their effectiveness.

2. Related Work

The subject of study of anomaly or novelty detection have been surveyed in a lot of different approaches. [Hodge and Austin 2004] provide a comprehensive study of classical techniques for identifying outliers using statistics and machine learning. An extensive review in cyber-security for detecting intruders in a computer network using deep learning is provided by [Kwon et al. 2017] Deep learning approaches also have been studied in fraud-detection problem by [Adewumi and Akinyelu 2017]. With image data, [Kiran et al. 2018] shows deep learning approaches, including a small review on GAN-based methods.

Although there are surveys that briefly cites GAN-based approaches, there is a lack of a comprehensive study on those methods, presenting their advantages and disadvantages and a methodological process for benchmarking. This work aims to fill that gap and brings a comprehensive reference for who wants to apply this technique on his own problem.

2.1. Generative Adversarial Network

The Generative Adversarial Networks [Goodfellow et al. 2014] work proposes a neural network training framework for estimating a generative model with a adversarial training. In the algorithm two neural networks are simultaneously trained, the generator network that learns to capture the data distribution during training, and outputs an artificial sample, and the discriminator that learns to identify inputs that comes from the original dataset or from the generator's output.

In this simultaneously training process, the generator receives a random gaussian noise vector also named as a latent vector, and outputs a matrix representing a new data point, then the discriminator receives fake samples produced by the generator and real samples from the original dataset for predict if the incoming sample is from the training data or not. This process can be visualized in the Figure 1.

The generator objective is to maximize the probability of the discriminator make a wrong prediction, and the discriminator objective is to identify correctly if a sample is original or artificially generated. This process is equal to a minimax game from game theory [Goodfellow et al. 2014], and the optimum situation is to find the Nash equilibrium, where the generator produces high quality data such that the discriminator can differentiate them with a probability of 50%.

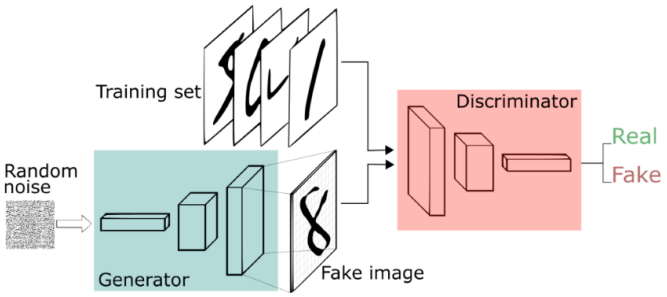

Figure 1. GAN illustration
Source: [Silva 2018]

3. Anomaly Detection with Generative Adversarial Networks

In this section is discussed each method proposed to be studied, showing how the GAN technique is applied for anomaly detection problems.

All of the methods studied relies on the generator network, that models the data distribution based on the input latent vector, which is a representation of an artificial sample in the latent space. In other words, the generator learns how to translate from the latent representation into an image. Changing values in this latent vector yields to modifying some aspect of the output. For example one position of the vector can represent the size of an object in the scene. The neural network is able to learn how to make these transitions. If we can gather this latent space representation for an existing original sample and the same process to an anomalous sample, we might expect some differences in the values of those vectors.

The original architecture of GAN does not provide a direct way to transform an image to the latent vector, only the other way. In the following subsections we will discuss each method and present how they transform this image to latent space as a latent vector.

3.1. AnoGAN

AnoGAN [Schlegl et al. 2017] is one of first methods introducing GANs for the anomaly detection task, which is based on a normal GAN training utilizing the Deep Convolutional Generative Adversarial Networks [Radford et al. 2015] architecture, using only non-anomaly samples during training. In test time each image sample needs to be mapped into latent space through an optimization process, which minimizes a loss function that is a weighted average of the similarity of the generated image and the original, and a discriminator score, that is calculated with the L_2 distance of an intermediate layer in the discriminator network. This average is the anomaly score utilized. A high score is expected to be an anomaly, and a low score is likely to be a non-anomalous sample. Since this work relies on an optimization problem on test time, it has a high computational cost.

3.2. Adversarially Learned Anomaly Detection

The Adversarially Learned Anomaly Detection (ALAD) [Zenati et al. 2018] paper uses the Bidirectional Generative Adversarial Networks (BiGANs) [Donahue et al. 2016] that enable the model to learn the mapping from a sample into a feature space, analogous to the latent space of standard GANs, this feature space is learned with autoencoders in the

BiGAN architecture. With this representation of features, the authors propose some methods for calculating the distance between samples in this representations, these equations are defined below:

$$L_1 = \|x - x'\|_1 \tag{1}$$

$$L_2 = \|x - x'\|_2 \tag{2}$$

In L_1 and L_2 the direct distance between the original sample x and the artificial sample x' after feed-forward in the generator is computed

$$Logits = \log\left(D_{xx}\left(x, x'\right)\right) \tag{3}$$

The $Logits$ is defined by the log of the output of the D_{xx} discriminator network.

$$Features = \|f_{xx}(x, x) - f_{xx}\left(x, x'\right)\|_1 \tag{4}$$

The $Features$ calculates the L_1 distance of the difference between the values of the feature layer f_{xx} presented in D_{xx} for both x and x'.

The authors elaborates that using the direct output of the discriminator D_{xx} will lead to random predictions, since in an optimal case, the encoder and decoder will capture the data distribution perfectly, in this case the discriminator will not be able to distinguish real and reconstructed samples.

Instead of using the direct output of the discriminator, the authors proposed the $Features$ measurement that gives the best anomaly score, showing that the features in the latent space yields important information about the data.

3.3. Fence GAN

In the work of Fence GAN [Phuc Ngo et al. 2019] the authors highlights that the standard GAN objective is to make the distribution of the generators output equal to the distribution of the original data, which is good for recreating new samples that are similar to the train data, but it is not sufficient for the anomaly detection problem.

The authors hypotheses that if during training the generators only produces samples in the boundary of the data distribution, the discriminator will acts as an anomaly classifier. To achieve that constraint, both generator and discriminator loss functions are modified.

$$\mathcal{L}^{FGAN}_{generator}\left(G_\theta, D_\phi, \mathcal{Z}\right) = EL + \beta \times DL \tag{5}$$

$$EL\left(G_\theta, D_\phi, \mathcal{Z}\right) = \frac{1}{N}\sum_{i=1}^{N}\left[\log\left(\left|\alpha - D_\phi\left(G_\theta\left(z_i\right)\right)\right|\right)\right] \tag{6}$$

$$DL\left(G_\theta, \mathcal{Z}\right) = \frac{1}{\frac{1}{N}\sum_{i=1}^{N}\left(\left\|G_\theta\left(z_i\right) - \mu\right\|_2\right)} \tag{7}$$

The generator loss function of FenceGAN is described by the Encirclement Loss EL and the Dispersion Loss DL. The Encirclement Loss will help the generator to produce data that lies on the boundary of the data distribution, based on the parameter α. The Dispersion Loss helps reinforce that the all generated points lies in the boundary of the data distribution.

With the generator producing data points that lies in the boundary of the data distribution, then the discriminator loss function needs to be modified too, because it will classify data points from the original data and data points that lies in the boundary.

$$\mathcal{L}_{discriminator}^{FGAN}\left(G_\theta, D_\phi, \mathcal{X}, \mathcal{Z}\right) = \frac{1}{N}\sum_{i=1}^{N}\left[-\log\left(D_\phi\left(x_i\right)\right) - \gamma\log\left(1 - D_\phi\left(G_\theta\left(z_i\right)\right)\right)\right] \tag{8}$$

To prioritize classifying the original data correctly, the hyper-parameter γ is introduced in the discriminator loss function. The value of γ ranges from [0, 1], with low values the function will produce high values when the discriminator does not classify real data points correctly, therefore it will learn the normal data distribution.

3.4. GANomaly

The GANomaly [Akcay et al. 2018] method is similar to ALAD [Zenati et al. 2018] approach, where the generator consists of encoder-decoder-encoder sub-networks, which results in a direct feature space representation.

For calculating the anomaly score the reconstruction and feature matching losses must be comined, which respectively calculates the difference of the input sample and the generated sample, and the difference between the sample in the feature space and the generated image, described by the following equation:

$$\mathcal{A}(\hat{x}) = \left\|G_E(\hat{x}) - E(G(\hat{x}))\right\|_1 \tag{9}$$

Where $\hat{x}$ is an incoming data sample, G_E is the first encoder inside the generator and E is the second encoder of the generator. Non-anomalous samples should yield into a value close to 0, and anomalies should have greater values.

3.5. OCGAN

The OCGAN [Perera et al. 2019] hypotheses that when training auto-encoders with multiple classes, a straight path in the latent space can generate different classes, therefore constraining the latent space to one specific class will produce high reconstruction errors when evaluating samples that are not from the target class. To constrain the latent space the authors propose a GAN architecture that uses two discriminators, a latent space network that outputs a probability of a sample come from the restricted latent space, and a visual discriminator that helps the model to generate high quality data without hurting the performance on the anomaly detection task.

For detecting anomalies a classifier network tries to capture how great the input image resembles content of the target class, so the input is the combination of the encoded image and the target class.

4. Experiment

For evaluating the selected methods, two datasets have been selected, the MNIST containing 70.000 of hand-written digits and CIFAR-10 with 60.000 of different objects, where both of them have 10 classes.

To working with the anomaly detection problem, each experiment used one class as the anomaly and the nine others as normal data, consequently having much more non-anomalous samples then the anomaly class. This process where executed for all classes in both datasets.

The Cifar-10 have equally distributed samples, therefore each experiment have 6000 training anomalous samples, 54000 training non-anomalous samples, 1000 anomalous test samples and 9000 non-anomalous test samples.

The data distribution of each experiment in the MNIST dataset can be visualized in Table 1.

	0	1	2	3	4	5	6	7	8	9
Training anomalies	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949
Training non-anomalies	54077	53258	54042	53869	54158	54579	54082	53735	54149	54051
Test anomalies	980	1135	1032	1010	982	892	958	1028	974	1009
Test non-anomalies	9020	8865	8968	8990	9018	9108	9042	8972	9026	8991

Table 1. Training and test data distribution for each class on MNIST dataset

For providing a better comparison in the results, the kernel density estimation (KDE) [Bishop 2006] method was selected as a baseline method, since it has shown great results in the anomaly detection task [Jan Latecki et al. 2007]. This method is based on estimation a probability density function of the training data, then incoming samples are evaluated if they lies in the data distribution modeled by KDE.

For each evaluated method the hyper-parameters used are the best ones reported by their authors. Each experiment where executed three times using different random seeds, and the final result is the average of those three executions. The implementation of ALAD[1], AnoGAN[2], FenceGAN[3] and GANomaly[4] is published at Github by their own authors. The results from OCGAN and KDE are gathered from [Perera et al. 2019].

We used the Area Under the Receiver Operating Characteristics (AUROC) [Bradley 1997] as the performance measurement for each method. The AUROC score measures how well a model can distinguish between two classes, the score varies from 0 (poor result) to 1 (best result).

[1]https://github.com/houssamzenati/Adversarially-Learned-Anomaly-Detection

[2]https://github.com/LeeDoYup/AnoGAN

[3]https://github.com/phuccuongngo99/Fence$_GAN$

[4]https://github.com/samet-akcay/ganomaly

5. Results

In the Table 2 the mean of the AUROC score for each experiment with the MNIST dataset can be observed, and also a final mean across all classes. For each experiment and the overall mean, the best values are highlighted.

In the MNIST results it is possible to visualize a big improvement from the classical KDE [Hodge and Austin 2004] method with the OCGAN [Perera et al. 2019] work, the other methods yields competitive results, with one exception with the ALAD method.

Method	0	1	2	3	4	5	6	7	8	9	Mean
KDE	0,855	0,996	0,710	0,693	0,844	0,776	0,861	0,844	0,669	0,825	0,835
ALAD	0,743	0,296	0,684	0,525	0,456	0,432	0,579	0,380	0,553	0,363	0,491
AnoGAN	0,966	0,992	0,850	0,887	0,894	0,883	0,947	0,935	0,849	0,924	0,909
FenceGAN	0,780	0,959	0,784	0,810	0,665	0,714	0,686	0,844	0,605	0,602	0,747
GANomaly	0,892	0,675	**0,946**	0,793	0,810	0,847	0,821	0,689	0,838	0,650	0,816
OCGAN	**0,998**	**0,999**	0,942	**0,963**	**0,975**	**0,980**	**0,991**	**0,981**	**0,939**	**0,981**	**0,981**

Table 2. Anomaly detection results for MNIST dataset

The CIFAR-10 experiment results can be observed in Table 3, which showed to be a more challenging task resulting in similar final scores, with the ALAD [Zenati et al. 2018] method presenting an increase of 3,3%, and none of the other method had a lower score than the KDE [Hodge and Austin 2004] method.

Method	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	Mean
KDE	0,658	0,520	**0,657**	0,497	0,727	0,496	0,758	0,564	0,680	0,540	0,611
ALAD	0,681	0,453	0,646	**0,641**	0,665	0,542	**0,771**	0,522	0,762	0,418	**0,644**
AnoGAN	0,671	0,547	0,529	0,545	0,651	0,603	0,585	0,625	0,758	**0,655**	0,614
FenceGAN	0,629	0,596	0,500	0,500	0,606	0,584	0,686	**0,647**	0,626	0,641	0,616
GANomaly	0,613	**0,627**	0,505	0,579	0,581	**0,623**	0,679	0,612	0,622	0,617	0,615
OCGAN	**0,757**	0,531	0,640	0,620	**0,723**	0,620	0,723	0,575	**0,820**	0,554	0,630

Table 3. Anomaly detection results for CIFAR-10 dataset

The results show that GAN-based methods can be a good approach for anomaly detection systems, the results showed that in both datasets GAN methods outperform the classical KDE [Hodge and Austin 2004] method.

6. Conclusion

This work presents recent efforts towards bringing better anomaly detection systems using Generative Adversarial Networks. The main challenge on this task is to transform the test data to the latent space for calculating an anomaly score. The AnoGAN [Schlegl et al. 2017] showed that this can be achieved through an optimization process in test time with a high computational cost, and later methods relying on autoencoders such as OCGAN [Perera et al. 2019] solved this problem at training time.

The latent space representation prove to be a good technique to represent contextual information of an image. The best results where from the methods that used this

representation to calculate an anomaly score. Since the KDE [Hodge and Austin 2004] method relies on estimating the data distribution based on the pixel values, it cannot model contextual information, therefore harming the algorithm performance.

Due to the promising results shown for the anomaly detection task, future work objective relies on testing GAN methods on datasets that have a higher resolution images. Another proposal is to use the generated samples for data augmentation in adversarial attack problem to increase the attackers power.

7. Acknowledgements

The authors gratefully acknowledge the financial support of University Center FACENS under the Scientific Initiation program, and to Ufscar Sorocaba for providing computational resources, accelerating the experiment execution.

References

Adewumi, A. O. and Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of System Assurance Engineering and Management*, 8(2):937–953.

Akcay, S., Atapour-Abarghouei, A., and Breckon, T. P. (2018). GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. *arXiv e-prints*, page arXiv:1805.06725.

Alguliyev, R., Aliguliyev, R., and Sukhostat, L. (2017). Anomaly detection in big data based on clustering. *Statistics, Optimization Information Computing*, 5.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30(7):1145–1159.

Brock, A., Donahue, J., and Simonyan, K. (2018). Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv e-prints*, page arXiv:1809.11096.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58.

Curtó, J. D., Zarza, H. C., and Kim, T. (2017). High-Resolution Deep Convolutional Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1711.06491.

Donahue, J., Krähenbühl, P., and Darrell, T. (2016). Adversarial Feature Learning. *arXiv e-prints*, page arXiv:1605.09782.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1406.2661.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126.

Jan Latecki, L., Lazarevic, A., and Pokrajac, D. (2007). Outlier detection with kernel density functions. pages 61–75.

Kiran, B. R., Thomas, D. M., and Parakkal, R. (2018). An overview of deep learning based methods for pervised and semi-supervised anomaly detection in videos. *arXiv e-prints*, page arXiv:1801.03149.

Kwon, D., Kim, H., Kim, J., C. Suh, S., Kim, I., and Kim, K. (2017). A survey of deep learning-based network anomaly detection. *Cluster Computing*.

Perera, P., Nallapati, R., and Xiang, B. (2019). OCGAN: One-class Novelty Detection Using GANs with Constrained Latent Representations. *arXiv e-prints*, page arXiv:1903.08550.

Phuc Ngo, C., Aristo Winarto, A., Kou Khor Li, C., Park, S., Akram, F., and Lee, H. K. (2019). Fence GAN: Towards Better Anomaly Detection. *arXiv e-prints*, page arXiv:1904.01209.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. page arXiv:1511.06434.

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. page arXiv:1703.05921.

Shetty, R., Fritz, M., and Schiele, B. (2018). Adversarial Scene Editing: Automatic Object Removal from Weak Supervision. *arXiv e-prints*, page arXiv:1806.01911.

Silva, T. (2018). An intuitive introduction to generative adversarial networks (gans).

Zenati, H., Romain, M., Foo, C. S., Lecouat, B., and Ramaseshan Chandrasekhar, V. (2018). Adversarially Learned Anomaly Detection. *arXiv e-prints*, page arXiv:1812.02288.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. (2018). A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. *arXiv e-prints*, page arXiv:1811.08055.