Anomaly detection Using Generative Adversarial Networks(GAN)- Survey

https://arxiv.org/pdf/1906.11632.pdf

Medium Article Link

- https://medium.com/@ashabalshiram.aher/anomaly-detection-using-generative-adversarial-networks-gan-ca433f2ac287

Introduction

- Anomaly detection is one of the most important problems across a range of domains, including manufacturing, medical imaging and cyber-security.
- Anomalies are patterns in data that do not conform to a well-defined notion of normal behavior.
- Anomaly detection is a significant problem faced in several research areas.
- Anomaly detection methods need to model the distribution of normal data, which can be complex and high-dimensional.

About GAN

- GANs
  - Generative adversarial networks (GANs) (Goodfellow et al., 2014) are one class of models that have been successfully used to model complex and high dimensional distributions.
- Conditional GANs
  - GANs can be extended to a conditional model (Mirza & Osindero, 2014) if both generator and discriminator are conditioned on some auxiliary information.
- BiGANs
  - Bidirectional GAN (Donahue et al., 2016) extends the GAN framework including an encoder that learns the inverse of the generator.

GANs for anomaly detection

- **Using AnoGAN Architecture:**
  - AnoGAN uses standard GAN, train only on positive samples, to learn a mapping from the latent space representation z to the realistic sample and uses this learned representation to map new, unseen, samples back to the latent space
- **Using EGBAD Architecture:**
  - EGBAD solves the AnonGAN disadvantages using Donahue et al. (2016) and Dumoulin et al. (2017) works that allows learning an encoder able to map input samples to their latent representation during the adversarial training.
- **Using GANomaly Architecture:**
  - train a generator network on normal samples to learn their manifold X while at the same time an autoencoder is trained to learn how to encode the images in their latent representation efficiently.
  - This Architecture requires below components as in a standard GAN architecture
    - Generator network
    - Discriminator network
    - Generator loss

Ablation Studies

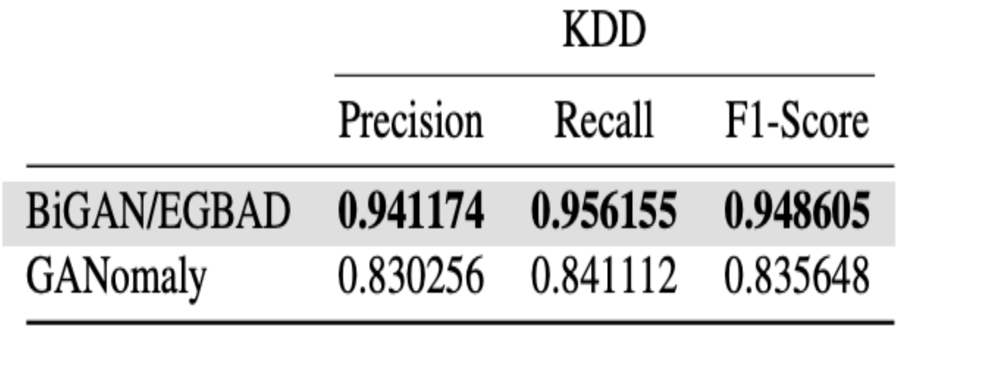
- in order to evaluate the performance of every Anomaly Detection algorithm described above, the reimplementation of all mentioned algorithms was done using in demand deep learning framework Tensorflow
- Datasets:
  - MNIST (28 × 28 pixels grayscale image in 10 classes)
  - Fashion MNIST (28 × 28 pixels grayscale images in 10 classes)
  - CIFAR-10 (32 × 32 color images in 10 classes)
  - KDD

Methodology

- All the mentioned datasets in ablation studies are taken together (train and test split). from this one big pool of examples, one class chose as an anomaly.
- after shuffling the dataset, training set is created by using 80% of the whole data while the remaining 20% is used for the testing set.
- this process is repeated for all the classes in the dataset.
- Each model is trained accordingly to its original implementation on the training set.
- Each model is tested on the whole dataset made up of both regular and anomalous data.

Results

- GAN networks were trained with different hyper-parameters configurations
- Below table states the results for KDD dataset



	KDD		
	Precision	Recall	F1-Score
BiGAN/EGBAD	**0.941174**	**0.956155**	**0.948605**
GANomaly	0.830256	0.841112	0.835648

Conclusions

- Implementation of algorithms and analysis allowed to verify effectiveness of the GANs based approach to anomaly detection problem
- also at the same time highlighted the difference between the original papers and the publicly available code.

References

- https://arxiv.org/pdf/1906.11632.pdf
- https://github.com/tSchlegl/f-AnoGAN
- https://arxiv.org/pdf/1411.1784.pdf
- https://openreview.net/pdf?id=S1EfylZ0Z
- https://arxiv.org/pdf/1805.06725.pdf
- https://developers.google.com/machine-learning/gan/generator
- https://developers.google.com/machine-learning/gan/discriminator
- https://arxiv.org/abs/1606.03498
- https://arxiv.org/abs/1802.06222
- http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf