Anomaly detection using GANs

Diane MT

10 juin 2021

1 Survey of Anomaly detection techniques

From [1] (2016).

Survey mainly on 2D data (i.e. not images) and most of the techniques mentioned are fairly old (some before 2000). SVM, CNN, tree-based methods and k-means (and derived like distance-based anomaly detection) look the most promising when applied to 3D data.

The main advantage of an anomaly detection is that there is almost no need for anomalous data compared to a standard object/feature detection technique. The classification based network techniques like SVM do not generalize well when noisy data is present in the training set, Robust SVM can be used to ignore noisy data. As for neural network, the downside is, as we know, the highly computational power required, at the time of the paper, RNN and neural network combined with statistical models were the frontrunners for anomaly detection.

Clustering and co-clustering based methods do not require pre-labeled data to extract rules, which is an advantage when those are limited in numbers. Regular clustering clusters the data considering the rows of the dataset whereas the co-clustering considers both rows and columns of the dataset simultaneously to produce clusters. As before, the downside are noisy data/images, noise can be considered anomalous. 2 assumptions when using clustering :

1. When a cluster contains both normal and anomalous data, it has been found that the normal data lie close to the nearest clusters centroid but anomalies are far away from centroids (Ahmed and Naser, 2013). Under this assumption,anomalous events are detected using a distance score.
2. In a clustering with clusters of various sizes, the smaller and sparser can be considered anomalous and the thicker normal. Instances belonging to clusters the sizes and/or densities below a threshold are considered anomalous.

The k-means approach (used by Münz et al. -2007) generate normal and anomalous clusters. Given the distances to the centroid, it is either classified as anomalous or normal and with a predefined threshold, if the distance between an instance and centroid is larger, the instance is treated as an anomaly. Other similar techniques consider the characteristics of the clusters (like compactness, separation, etc..) to distinguish normal to abnormal data.

Co-clustering on the other hand, defines a clustering criterion and the optimizes it. It has the benefits of providing 'compressed' representation and preserving information contained in the original dataset, and is significantly less computationally complex than traditional k-means.

2 Survey on GANs for anomaly detection

GAN : Generative Adversarial Network, ML frameworks designed by Goodfellow et al. (2014), given a training set, this network learns to generate new data with the same characteristics/statistics. Applied to anomaly detection, the generated image is compared to the input, if it derives too much then it can be considered as anomalous data/image.

From [3] (2019).

Basic GANs are a framework in which 2 models, a discriminator D and a generator G are trained simultaneously. G captures the data distribution while D estimates the probability that a sample came from the training set rather than the generator.

Conditional GANs (or CGANs), extend the basic framework by conditioning G or D on some extra information (e.g. class labels). Bidirectional GAN (or BiGAN), extends the original framework by including an encoder that learns the inverse of the generator (i.e. $E = G^{-1}$), where the encoder is a non linear parametric function (like G and D), and can be trained using gradient descent.

GANs specifically for anomaly detection started flourishing in 2017, the first instance being AnoGAN.

The former, however faces performances issues as, for every new input, it requires Γ(backpropagation) optimization steps for every new input ; meaning a consequent test time. Efficient GAN-based Anomaly Detection (EGBAD) tries to solve the AnoGan testing problem by utilizing a BiGAN architecture. They succeed as EGBAD allow computing the anomaly scores without the Γ steps for each input. The anomaly scores are then used to determine a threshold, commonly during the validation phase.

GANomaly (2018) take its inspiration from the last 3 GANs mentioned. GANomaly uses a discriminator network and a generator network consisting in 3 elements : encoder (G_E), decoder (G_D) and a final encoder (G_E). Both encoders have the same architecture, G_E takes an image as an input and outputs an encoded version, while the discriminator takes an image and output a reconstructed version of it. The discriminator D is the same as basic GAN architectures ; they also introduce a generator loss, which is the weighted sum of 3 losses, respectively : adversarial loss, contextual loss ans encoder loss. Finally, the anomaly score is easy to interpret as they compute it for every sample in the test set, and they apply feature scaling to have a set of individual anomaly scores within the the probabilistic range.

Next : the 3 GAN models that dominate the field in anomaly detection (using GANs).

3 GANomaly : Semi-Supervised Anomaly Detection

From [2] (2018).

Introduced to tackle the lack of labelled datasets, especially for unusual classes, GANomaly is trained only using 'normal' images/data, and effectively is a semi-supervised method. As thoroughly explained in the paper, the anomaly detection main problem is to 'learn' the data distribution (say p_X) of the normal sample during training and identifying abnormal ones during test times. Commonly via an Anomaly Score ($A(x)$), which can be defined in plenty of ways

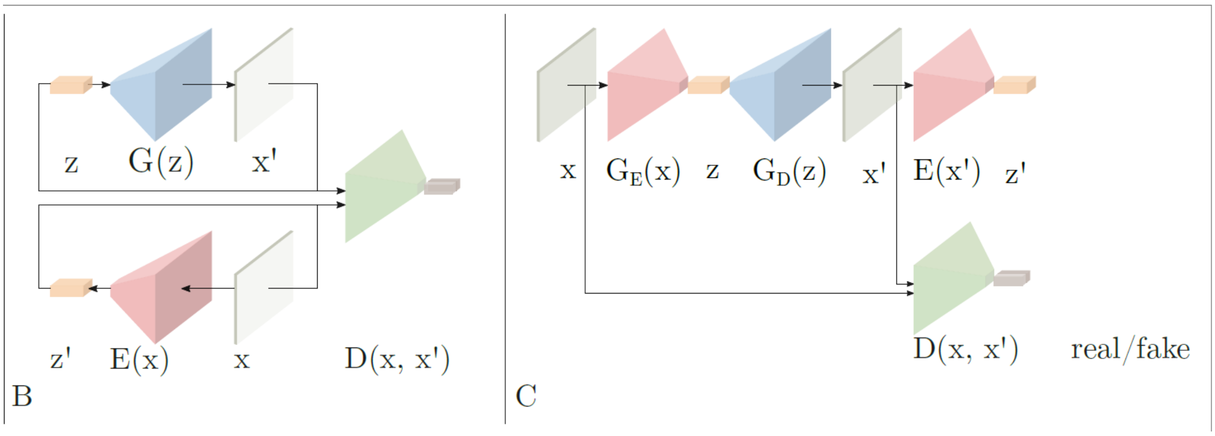


FIGURE 1 – Comparison of 2 frameworks. B : EGBAD pipeline, C : GANomaly pipeline

given a specific model/training process.

Personal note : given the importance of the distribution of normal data, the initializer is crucial ; the same can be said about the assumption that normal data are identically (or almost identically) distributed. Noisy data as said before are a relative threat to the stability of training/outcome of a given GAN (for anomaly) model. This kind of method should be suited as the type of data we aim to use have more or less the same characteristics.

GANomaly is defined as a standard BiGAN architecture with a generator G, en encoder E and a disciminator D. However, it uses an adversarial autoencoder within an encoder-decoder-encoder framework (acting as the generator), which captures the (training) data distribution jointly with the latent vector space.

The objective function of GANomaly is the weighted sum of the contextual (L_1 distance), adversarial and encoder (L_2 distances) losses. They define their anomaly score as $A(\hat{x}) = ||G_E(\hat{x}) - E(G(\hat{x}))||_1$ which is equivalent to the encoder loss for a test sample $\hat{x}$ (with L_1 distance). The anomaly score is computed for every sample $\hat{x}$, then a feature scaling is applied to range the scores within a $[0, 1]$ range. As before, a larger $A(\hat{x})$ indicates an anomalous data/image.

On the simplistic MNIST dataset they achieve better result than EGBAD aswell as for the CIFAR-10 one.

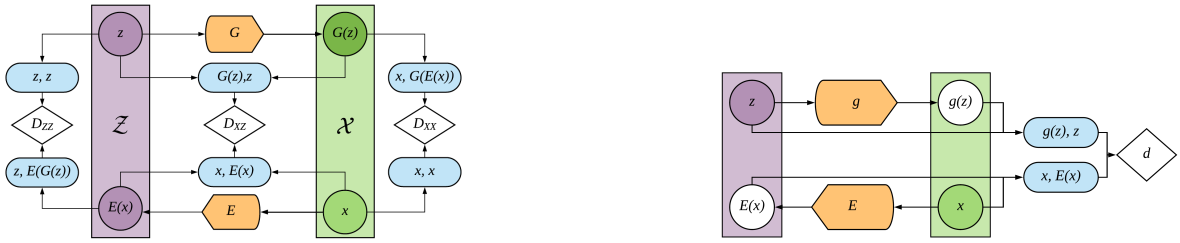
4 EGBAD : Efficient GAN-Based Anomaly Detection

From [4] (2018).

EGBAD is used for both image (e.g. CIFAR-10) and tabular dataset (e.g. KDD99), with both approaches they either match or best other methods (at the time). Based on BiGAN, EGBAD simultaneously learns an encoder E (that maps an input sample x to a latent representation z) and a generator G, such that $E = G^{-1}$, and finally the models learns a discriminator D, all that during the training phase. Here, D considers either a generated input or an input from the encoder (i.e. it takes into account the latent representation).

They use the following anomaly score :

$$A(x) = \alpha L_G(x) + (1 - \alpha) L_D(x)$$



(a) ALAD training architecture

(b) EGBAD training architecture

Figure 2 – ALAD & EGBAD architectures comparison

Where x is the input image/data L_G is the reconstruction loss defined by $L_G(x) = ||x - G(E(x))||_1$, and $L_D(x)$ the discriminator-based loss by $L_D(x) = ||f_D(x, E(x)) - f_D(G(E(x)), E(x))||_1$ (f_D returns the layer preceding the logits for the given inputs in the discriminator).

Thus, samples with large values of $A(x)$ are said to be more likely to be anomalous.

5 ALAD : Adversarially Learned Anomaly Detection

From [5]

ALAD authors are the same as the EGBAD model, but they do not compare both models in the original paper.

The core ideas are the same as before, however, with ALAD they introduce techniques to stabilize the training process, namely the optimization process is much more detailed in ALAD. They regularize the conditional distributions by adding a conditional entropy constraint and then apply spectral normalization. In the end, ALAD is formed by 3 learned discriminator : D_{xx}, D_{xz} and D_{zz}, where x is the *real* input and z the latent representation. The optimization problem solved during training is now

$$min_{G,E} max_{D_{xx},D_{xz},D_{zz}} (V(D_{xx}, E, G) + V(D_{xz}, E, G) + V(D_{zz}, E, G))$$

.

For comparison, the one solved during training of the EGBAD model is just

$$min_{G,E} max_D (V(D, E, G))$$

.

The anomaly score is now based on the L_1 reconstruction error in the feature space (xx). Formally : $A(x) = ||f_{xx}(x, x) - f_{xx}(x, G(E(x)))||_1$, where $f(.,.)$ are the activations of the layers in D_{xx} given a pair of inputs.

Références

[1] M. Ahmed, A. Naser Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60 :19–31, 2016.

[2] S. Akcay, A. A. Abarghouei, and T. P. Breckon. Ganomaly : Semi-supervised anomaly detection via adversarial training. *CoRR*, abs/1805.06725, 2018.

[3] F. D. Mattia, P. Galeone, M. D. Simoni, and E. Ghelfi. A survey on gans for anomaly detection. *CoRR*, abs/1906.11632, 2019.

[4] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. Efficient gan-based anomaly detection. *CoRR*, abs/1802.06222, 2018.

[5] H. Zenati, M. Romain, C. S. Foo, B. Lecouat, and V. R. Chandrasekhar. Adversarially learned anomaly detection. *CoRR*, abs/1812.02288, 2018.