**Abstract**

The traditional approaches such as Decision Trees, SVM are used to classify based on the intrusion and based on availability of the data preference of GAN is increased as such to work with the data even though abnormalities.

**1.Algorithm Description**

1.1 Introduction

As relatively the data with intrusion is dominantly less, the effective learning of the model is more dependant. So, the generative approach gives more effective results. An observation for the Real space and Latent space is concerned.

The essence of using GAN for anomaly detection is learning the feature of normal data and accurately find the ‘normal version’ of

the testing sample. AnoGAN is proposed to better extract the

feature of normal samples, by establishing a mapping between the

real space and latent space.And through the discrepancy between

the mapped sample and testing sample,we can approach the optimal

‘normal version’ of testing sample.But this mapping is based on

the back-propagation algorithm, thus when the dimension of data

increases, this model will be time-consuming and not suitable for

timely intrusion detection.

To reduce the time complexity a new approach based on BiGAN architecture is developed using redefined loss function other than cross entropy.

**1.1.2 Variants of GAN:**

|  |  |
| --- | --- |
| **GAN Type** | **Objective Function** |
| GAN | The original (JSD divergence) |
| WGAN | EM distance objective |
| Improved WGAN | No weight clipping on WGAN |
| LSGAN | L2 loss objective |
| RWGAN | Relaxed WGAN framework |
| McGAN | Mean/covariance minimization objective |
| GMMN | Maximum mean discrepancy objective |
| MMD GAN | Adversarial kernel to GMMN |
| Cramer GAN | Cramer distance |
| Fisher GAN | Chi-square objective |
| EBGAN | Autoencoder instead of discriminator |
| BEGAN | WGAN and EBGAN merged objectives |
| MAGAN | Dynamic margin on hinge loss from EBGAN |

For the generative model, we have different type of GANs which can be implemented. Among these, a simple JSD based objective function would result be efficient according to mathematics reasoning.

**1.2 Mathematical Basis of the Algorithm**

**1.2.1 Using min-max objective function:**

Different strategies had been explored to optimize the training

process of GAN. The most widely used goal is a minimax objective

which illustrated as below:

min GmaxDV(D,G)

V (D,G) = Ex∼pX [logD(x)] + Ez∼pZ [log(1 − D(G(z)))]

In the goal above where cross-entropy loss function using log criterion is a classic tactic.

**1.2.2 Using Shannon Entropy**

Cross-entropy can be used to measure the Shannon Entropy needed to eliminate the uncertainty between two distributions, so it should naturally be the measurement of the disparity between two distributions P and Q:

H(P||Q) = Ex∼pX [− logQ(x)] = − sum(P(x) logQ(x))

Since during the training process of GAN, P can be seen as a constant

variable which represents the distribution of normal sample.

**1.2.3 Using KL divergence:**

Now, Kullback-Leibler(KL) divergence also can be used to weigh

the diversity of two distributions because:

KL(P ∥Q) = H(P ∥Q) − H(P)

Yet there is a deficiency in KL divergence: KL(P ∥Q) , KL(Q∥P),

which means the KL divergence is asymmetrical so it can not be

used to represent the distance between two distributions. This a

fatal factor since inconsistent discrepancy brings no benefit to training

Thus, Jensen-Shannon(JS) divergence is designed as follows to satisfy the symmetry required by distance:

JS(P||Q) =1/2 KL(P ||(P + Q)/2 ) +1/2 KL(Q||(P + Q)/2 )

Under the sway of dimension expansion, two distributions

will have few overlapping inevitably, but discrete feature will aggravate it because of One-Hot Encoding or Dummy Encoding. And

when two distribution have few overlapping, JS divergence will

unavoidably converge to a constant, then leads to the happening of

vanishing gradient. Fortunately, in previous work, Martin Arjovsky

et al.[1] supplanted the JS divergence with Wasserstein Distance,

which performs well even on the discrete distribution. Inspired by

Wasserstein distance, we modify the training goal of our model as follows:

V (D, E,G) = Ex∼pX [Ez∼pE(・ |x) ∥D(x, z)∥w]

+ Ez∼pZ [Ex∼pG(・ |z)[1 − ∥D(x, z)∥w]]

**1.3 Model Architecture**

Using AlexNet and BiGAN architectures as reference, we observe that the BiGAN can give better solutions with the generative nature of the network. It has 5 CNN layers and 3 Max Pool layers this converts 227 X 227 data entry to 6 X 6 and this reverts back the latent space dataset to the original valued dataset to compare as a GAN.

In the same way, with respect to Generator, the neural network is the reverse sequence of the network. This produces the synthetic data required and manipulates.

**1.4 Implementation**