Adversarially Learned Anomaly Detection

Houssam Zenati*†¶, Manon Romain*‡¶, Chuan-Sheng Foo*¶, Bruno Lecouat§¶ and Vijay Chandrasekhar¶
†*CentraleSupélec, houssam.zenati@student.ecp.fr*
‡*École Polytechnique, manon.romain@polytechnique.edu*
§*Télécom ParisTech*
¶*Institute for Infocomm Research, A*STAR, {foocs,bruno_lecouat,vijay}@i2r.a-star.edu.sg*
**Authors contributed equally to this work*

Anomaly Detection II

Adversarially Learned Anomaly Detection

Henrik Strangalies

Institute of Computer Science
Freie Universität Berlin

Overview

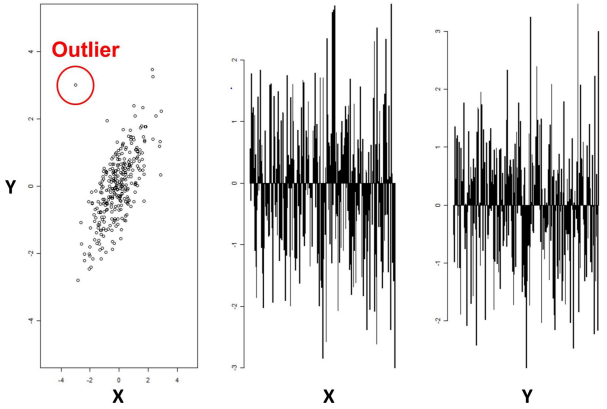
Table of Contents

Introduction to Anomaly Detection

Definition

Anomaly detection is the identification of rare items, events or observations by differing significantly from the majority.

Outliers



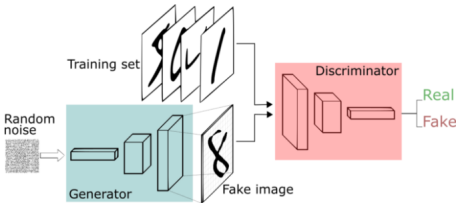
https://towardsdatascience.com/how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7

Applications of Anomaly Detection for Cybersecurity

- Detection of unauthorized access attempts (Intrusion detection).
- Detection of suspicious activity e.g. unusual types of requests.
- Prevent sensitive data leaks.

Table of Contents

Recap: Structure of GANs



- Generator maps random variables sampled from a latent distribution (typically Gaussian) to the input data space.
- Discriminator tries to distinguish real data samples from samples produced by Generator.
- Generator and discriminator are in competition whether generator can better fool the discriminator or the discriminator can better detect between real or fake image.

Anomaly Detection Task

How to **detect anomalies** from GANs?

Anomaly Detection Task

How to **detect anomalies** from GANs?

Assumption

Being able to learn the distribution of the normal data is key for the anomaly detection task.

Anomaly Detection Task

Two approaches for **Anomaly Detection** using GANs:

1. Use sampling to estimate the likelihood of an input and determine if it is an outlier.
2. "Invert" the generator to find latent variables that minimize reconstruction error.

Anomaly Detection Task

Two approaches for **Anomaly Detection** using GANs:

1. Use sampling to estimate the likelihood of an input and determine if it is an outlier.
2. "Invert" the generator to find latent variables that minimize reconstruction error.

- Both approaches are computationally expensive.
- It is neither possible to explicitly compute a likelihood nor obtain the latent representation for a given data sample directly using the vanilla GAN.

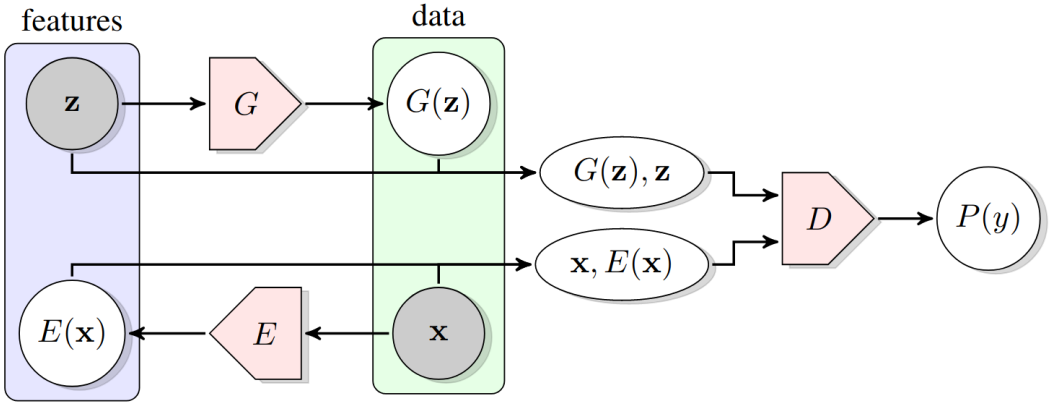
Anomaly Detection Task

Two approaches for **Anomaly Detection** using GANs:

1. Use sampling to estimate the likelihood of an input and determine if it is an outlier.
2. "Invert" the generator to find latent variables that minimize reconstruction error.

- Both approaches are computationally expensive.
- It is neither possible to explicitly compute a likelihood nor obtain the latent representation for a given data sample directly using the vanilla GAN.

Structure of BiGANs



BiGANs

- Encoder should learn to invert the generator where both are not directly connected. They never see their outputs.
- The BiGAN encoder learns to predict **latent representations**.
- It has been demonstrated that **these representations** capture attributes of the data which can be considered as labels.

BiGAN for Anomaly Detection Task

How to **detect anomalies** with BiGAN?

BiGAN for Anomaly Detection Task

How to **detect anomalies** with BiGAN?

Assumption

Normal samples should be accurately reconstructed whereas anomalous samples will likely be poorly reconstructed.

BiGAN for Anomaly Detection Task

How to **detect anomalies** with BiGAN?

Assumption

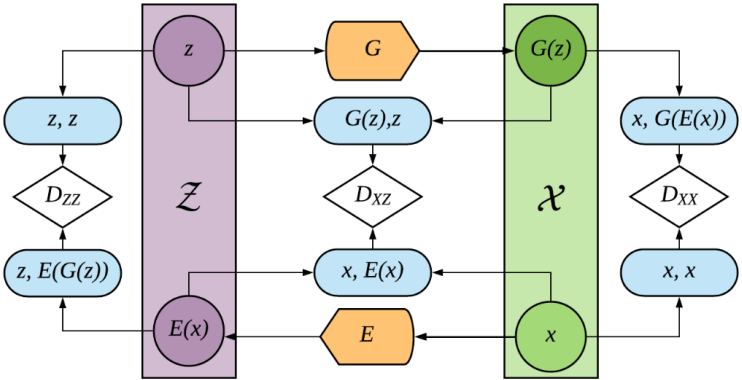
Normal samples should be accurately reconstructed whereas anomalous samples will likely be poorly reconstructed.

$\Rightarrow$ Reconstruction is done by: $G(E(x)) \approx x$.

Problem of BiGAN for Anomaly Detection Task

- While in theory the joint distributions $p_G(x, z) = p_Z(z)p_G(x|z)$ and $p_E(x, z) = p_X(x)p_E(z|x)$ will be identical, in practice this is often not the case as training does not necessarily converge to a solution of the saddle-point problem.
- This potentially results in a violation of cycle-consistency so $G(E(x)) \not\approx x$
  $\Rightarrow$ Such a violation would create issues for reconstruction-based anomaly detection methods.

Adversarially Learned Anomaly Detection (ALAD)



Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

Known from the **BiGAN**:

$$V(D_{xz}, E, G) = \mathbb{E}_{x \sim p_x}[logD_{xz}(x, E(x))] + \mathbb{E}_{z \sim p_z}[1 - logD_{xz}(G(z), z)]$$

Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

Known from the **ALICE framework**:

$$V(D_{xx}, E, G) = \mathbb{E}_{x \sim p_x}[logD_{xx}(x, x)] + \mathbb{E}_{x \sim p_x}[1 - logD_{xx}(x, G(E(x)))]$$

Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

For stabilizing GAN training, the **ALAD** Paper added another conditional entropy constraint:

$$V(D_{zz}, E, G) = \mathbb{E}_{z \sim p_z}[logD_{zz}(z, z)] + \mathbb{E}_{z \sim p_z}[1 - logD_{zz}(E(G(z)), z)]$$

Objective of ALAD

The **Objective of ALAD** is defined as: $min_{G,E} max_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$ where

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

where

$$V(D_{xz}, E, G) = \mathbb{E}_{x \sim p_x}[logD_{xz}(x, E(x))] + \mathbb{E}_{z \sim p_z}[1 - logD_{xz}(G(z), z)]$$

$$V(D_{xx}, E, G) = \mathbb{E}_{x \sim p_x}[logD_{xx}(x, x)] + \mathbb{E}_{x \sim p_x}[1 - logD_{xx}(x, G(E(x)))]$$

$$V(D_{zz}, E, G) = \mathbb{E}_{z \sim p_z}[logD_{zz}(z, z)] + \mathbb{E}_{z \sim p_z}[1 - logD_{zz}(E(G(z)), z)]$$

ALAD for Anomaly Task

How to **detect anomalies** with ALAD?

ALAD for Anomaly Task

How to **detect anomalies** with ALAD?

Assumption

Normal samples should be accurately reconstructed whereas anomalous samples will likely be poorly reconstructed.

$\Rightarrow$ Reconstruction is done by: $G(E(x)) \approx x$.

ALAD for Anomaly Task

- It is necessary to **learn the manifold of the data** so as to recover precise latent representations that result in accurate reconstructions of normal samples.
- The other key component of ALAD is an **anomaly score** that quantifies the distance between the original samples and their reconstructions.

Anomaly Score Function

A **score function** A(x) based on the L_1 reconstruction error in the feature space is defined as:

$$A(x) = ||f_{xx}(x, x) - f_{xx}(x, G(E(x)))||_1$$

where $f(.,.)$ denotes the activations of the layer before the logits in the D_{xx} network for the given pair of input sample.

Anomaly Score Function

A **score function** A(x) based on the L_1 reconstruction error in the feature space is defined as:

$$A(x) = ||f_{xx}(x, x) - f_{xx}(x, G(E(x)))||_1$$

where $f(.,.)$ denotes the activations of the layer before the logits in the D_{xx} network for the given pair of input sample.

Corollary

Samples with larger values of A(x) are deemed more likely to be anomalous.

ALAD Algorithm

Algorithm 1 Adversarially Learned Anomaly Detection

Input $\boldsymbol{x}, \sim p_{\mathcal{X}_{Test}}(x), E, G, f_{xx}$ where f_{xx} is the feature layer of D_{xx}

Output $A(\boldsymbol{x})$, where A is the anomaly score

procedure INFERENCE
- $\tilde{\boldsymbol{z}} \leftarrow E(\boldsymbol{x})$ ▷ Encode samples
- $\hat{\boldsymbol{x}} \leftarrow G(\tilde{\boldsymbol{z}})$, ▷ Reconstruct samples
- $f_\delta \leftarrow f_{xx}(\boldsymbol{x}, \hat{\boldsymbol{x}})$
- $f_\alpha \leftarrow f_{xx}(\boldsymbol{x}, \boldsymbol{x})$
- return $\|f_\delta - f_\alpha\|_1$

end procedure

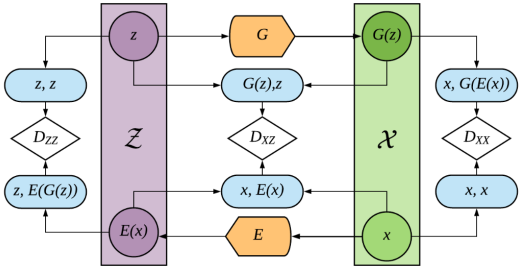


Table of Contents

Experiment setup

- For each dataset, 80% of the whole official dataset is used for **training** and the remaining 20% as are kept as **test set**.
- 25% from the training set are taken for a validation set.
- **Anomalous samples** from both training and validation sets are discarded.

Tabular dataset: KDDCup99

The KDDCup99 is a **network intrusion** dataset.

feature name	*description*	*type*
hot	number of ``hot'' indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of ``compromised'' conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if ``su root'' command attempted; 0 otherwise	discrete
num_root	number of ``root'' accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise	discrete
is_guest_login	1 if the login is a ``guest''login; 0 otherwise	discrete

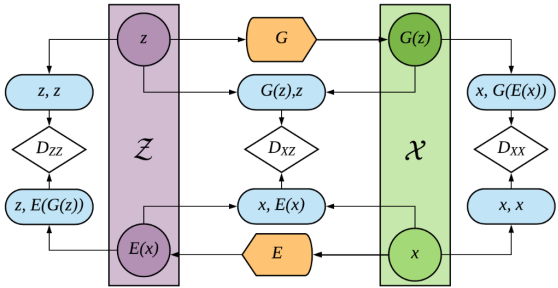
Data setup: KDDCup99

- The dataset contains samples of 41 dimensions, where 34 of them are continuous and 7 are categorical.
- **Preprocessing**: For categorical features, one-hot representation are used to encode them resulting in a total of **121 features**.
- Due to the high proportion of outliers in the KDD dataset, **"normal"** data samples are treated as **anomalies**.
- 20% of samples with the highest anomaly scores A(x) are classified as anomalies (positive class).
- Used metrics: Precision, Recall, F1 score.

Architecture and Hyper-parameters: KDDCup99

Operation	Units	Non Linearity	Batch Norm.	Dropout
$\boldsymbol{E(x)}$				
Dense	64	LReLU	×	0.0
Dense	1	None	×	0.0
$\boldsymbol{G(z)}$				
Dense	64	ReLU	×	0.0
Dense	128	ReLU	×	0.0
Dense	121	None	×	0.0
$\boldsymbol{D_{\mathbf{xz}}(x,z)}$				
Only on x				
Dense	128	LReLU	√	0.0
Only on z				
Dense	128	LReLU	×	0.5
Concatenate outputs				
Dense	128	LReLU	×	0.5
Dense	1	Sigmoid	×	0.0
$\boldsymbol{D_{\mathbf{xx}}(x,x')}$				
Concatenate x and x'				
Dense	128	LReLU	×	0.2
Dense	1	Sigmoid	×	0.0
$\boldsymbol{D_{\mathbf{zz}}(z,z')}$				
Concatenate z and z'				
Dense	32	LReLU	×	0.2
Dense	1	Sigmoid	×	0.0
Optimizer	Adam($\alpha = 10^{-5}$, $\beta_1 = 0.5$)			
Batch size	50			
Latent dimension	32			
Epochs	100			
LReLU slope	0.2			
Weight, bias init.	Xavier Initializer, Constant(0)			

Table XI
KDD99 ALAD ARCHITECTURE AND HYPERPARAMETERS



Binary Cross Entropy with Logits

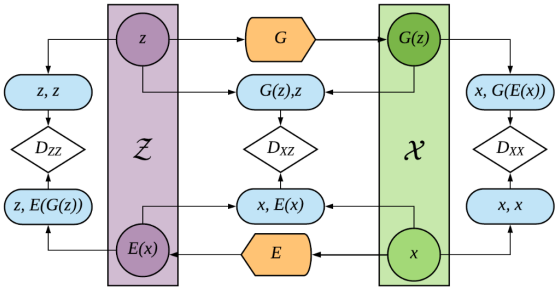
The **Binary Cross Entropy with Logits** is defined as:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i * log(\sigma(p(y_i))) + (1 - y_i) * log(1 - \sigma(p(y_i)))$$

- Predictions can be just binary, i.e. belonging to positive class or negative one.
- When the observation belongs to positive class the first term of the formula becomes active and the second part vanishes (vice versa in the other case).

Ablation study: With(-out) spectral normalization (SN) or latent Discriminator (DL)

Model	Precision	Recall	F1 score
	KDD99		
Baseline	0.948±0.007	0.963±0.007	0.955±0.007
Baseline + DL	0.944±0.008	0.959±0.008	0.951±0.008
Baseline + SN	0.942±0.004	0.957±0.004	0.949±0.004
Baseline + SN + DL	0.943±0.002	0.958±0.002	0.950±0.002



Ablation study: Various anomaly score functions

$$
\begin{aligned}
L_1 \quad &: A(x) = ||x - x'||_1 \\
L_2 \quad &: A(x) = ||x - x'||_2 \\
\text{Logits} \quad &: A(x) = \log(D_{xx}(x, x')) \\
\text{Features} \quad &: A(x) = ||f_{xx}(x, x) - f_{xx}(x, x')||_1
\end{aligned}
$$

Model	Precision	Recall	F1 score
	KDD99		
L_1	0.9113 ± 0.0627	0.9258 ± 0.0637	0.9185 ± 0.0632
L_2	0.9316 ± 0.0155	0.9464 ± 0.0157	0.9389 ± 0.0156
Logits	0.9221 ± 0.0172	0.9368 ± 0.0174	0.9294 ± 0.0173
Features	**0.9427 ± 0.0018**	**0.9577 ± 0.0018**	**0.9501 ± 0.0018**

Operation	Units	Non Linearity	Batch Norm.	Dropout
$\boldsymbol{E(x)}$				
Dense	64	LReLU	×	0.0
Dense	1	None	×	0.0
$\boldsymbol{G(z)}$				
Dense	64	ReLU	×	0.0
Dense	128	ReLU	×	0.0
Dense	121	None	×	0.0
$\boldsymbol{D_{xz}(x, z)}$				
Only on x				
Dense	128	LReLU	✓	0.0
Only on z				
Dense	128	LReLU	×	0.5
Concatenate outputs				
Dense	128	LReLU	×	0.5
Dense	1	Sigmoid	×	0.0
$\boldsymbol{D_{xx}(x, x')}$				
Concatenate x and x'				
Dense	128	LReLU	×	0.2
Dense	1	Sigmoid	×	0.0
$\boldsymbol{D_{zz}(z, z')}$				
Concatenate z and z'				
Dense	32	LReLU	×	0.2
Dense	1	Sigmoid	×	0.0
Optimizer	Adam($\alpha = 10^{-5}$, $\beta_1 = 0.5$)			
Batch size	50			
Latent dimension	32			
Epochs	100			
LReLU slope	0.2			
Weight, bias init.	Xavier Initializer, Constant(0)			

Table XI
KDD99 ALAD ARCHITECTURE AND HYPERPARAMETERS

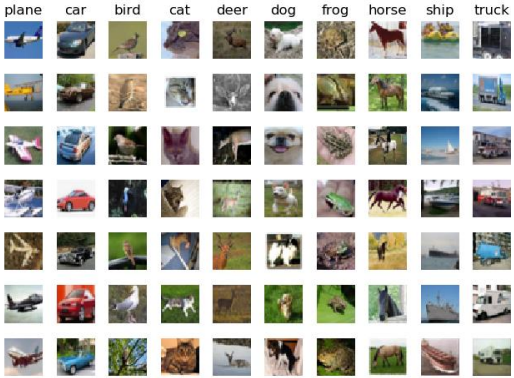
Performance on tabular data

PERFORMANCE ON TABULAR DATASETS

Dataset	Model	Precision	Recall	F1 score
KDD99	OC-SVM	0.7457	0.8523	0.7954
	IF	0.9216	0.9373	0.9294
	DSEBM-r	0.8521	0.6472	0.7328
	DSEBM-e	0.8619	0.6446	0.7399
	DAGMM	0.9297	0.9442	0.9369
	AnoGAN	0.8786	0.8297	0.8865
	ALAD	**0.9427**	**0.9577**	**0.9501**

Image dataset: CIFAR-10

The CIFAR-10 dataset contains **10 classes** of images.



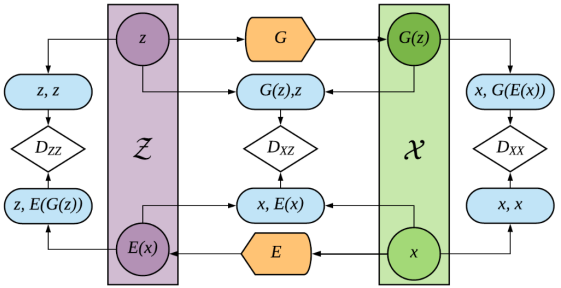
Experiment Setup

- **Preprocessing:** Pixels were scaled to be in range [-1,1].
- Ten different datasets each CIFAR-10 are computed where one class is deemed to be **normal** and considering images from the remaining 9 as **anomalous**.
- Used metrics: AUROC.

Architecture and Hyper-parameters: CIFAR-10

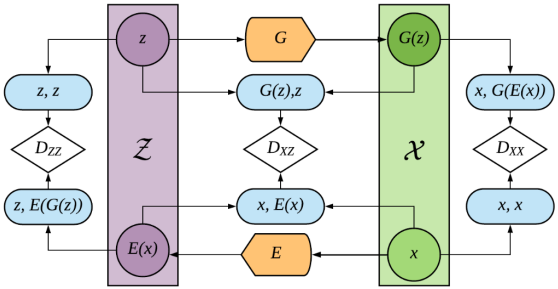
Operation	Kernel	Strides	Filters Units	Non Linearity	Batch Norm.
$\boldsymbol{E(x)}$					
Conv2D	4×4	2×2	128	LReLU	$\surd$
Conv2D	4×4	2×2	256	LReLU	$\surd$
Conv2D	4×4	2×2	512	LReLU	$\surd$
Conv2D*	4×4	1×1	100	None	$\times$
$\boldsymbol{G(z)}$					
Trans. Conv2D*	4×4	2×2	512	ReLU	$\surd$
Trans. Conv2D	4×4	2×2	256	ReLU	$\surd$
Trans. Conv2D	4×4	2×2	128	ReLU	$\surd$
Trans. Conv2D	4×4	2×2	3	Tanh	$\surd$
$\boldsymbol{D_{xx}(x,x)}$					
Only on x					
Conv2D	4×4	2×2	128	LReLU	$\times$
Conv2D	4×4	2×2	256	LReLU	$\surd$
Conv2D	4×4	2×2	512	LReLU	$\surd$
Only on z					
Conv2D†	4×4	2×2	512	LReLU	$\times$
Conv2D†	4×4	2×2	512	LReLU	$\times$
Concatenate outputs					
Conv2D†	1×1	1×1	1024	LReLU	$\times$
Conv2D	1×1	1×1	1	LReLU	$\times$
$\boldsymbol{D_{zz}(z,z')}$					
Concatenate x and x'					
Conv2D†	5×5	2×2	64	LReLU	$\times$
Conv2D†	5×5	2×2	128	LReLU	$\times$
Dense			1	None	$\times$
$\boldsymbol{D_{xz}(x,z')}$					
Concatenate z and z'					
Dense†			64	LReLU	$\times$
Dense†			32	LReLU	$\times$
Dense†			1	LReLU	$\times$
Optimizer	Adam($\alpha = 2 * 10^{-4}, \beta_1 = 0.5$)				
Batch size	32				
Latent dimension	100				
Max Epochs	100				
Patience	10				
LReLU slope	0.2				
Weight & bias initialization	Isotropic gaussian ($\mu = 0$, $\sigma = 0.01$) Constant(0)				

Table IX
CIFAR-10 AND SVHN ALAD ARCHITECTURE AND HYPERPARAMETERS



Ablation study: With(-out) spectral normalization (SN) or latent Discriminator(DL)

CIFAR-10	
Baseline	0.5701 ± 0.1282
Baseline + DL	0.5361 ± 0.1348
Baseline + SN	0.5991 ± 0.1308
Baseline + SN + DL	0.6072 ± 0.1201



Ablation study: Various anomaly score functions

$$
\begin{aligned}
L_1 \quad &: A(x) = ||x - x'||_1 \\
L_2 \quad &: A(x) = ||x - x'||_2 \\
\text{Logits} \quad &: A(x) = \log(D_{xx}(x, x')) \\
\text{Features} \quad &: A(x) = ||f_{xx}(x, x) - f_{xx}(x, x')||_1
\end{aligned}
$$

CIFAR-10	
L_1	0.6066 ± 0.1006
L_2	0.6012 ± 0.1088
Logits	0.5396 ± 0.0783
Features	0.6072 ± 0.1201

Operation	Kernel	Strides	Filters Units	Non Linearity	Batch Norm.
$\boldsymbol{E(x)}$					
Conv2D	4×4	2×2	128	LReLU	$\surd$
Conv2D	4×4	2×2	256	LReLU	$\surd$
Conv2D	4×4	2×2	512	LReLU	$\surd$
Conv2D*	4×4	1×1	100	None	$\times$
$\boldsymbol{G(z)}$					
Trans. Conv2D*	4×4	2×2	512	ReLU	$\surd$
Trans. Conv2D	4×4	2×2	256	ReLU	$\surd$
Trans. Conv2D	4×4	2×2	128	ReLU	$\surd$
Trans. Conv2D	4×4	2×2	3	Tanh	$\surd$
$\boldsymbol{D_{xx}(x, x)}$					
Only on x					
Conv2D	4×4	2×2	128	LReLU	$\times$
Conv2D	4×4	2×2	256	LReLU	$\surd$
Conv2D	4×4	2×2	512	LReLU	$\surd$
Only on z					
Conv2D†	4×4	2×2	512	LReLU	$\times$
Conv2D†	4×4	2×2	512	LReLU	$\times$
Concatenate outputs					
Conv2D†	1×1	1×1	1024	LReLU	$\times$
Conv2D	1×1	1×1	1	LReLU	$\times$
$\boldsymbol{D_{xx}(x, x')}$					
Concatenate x and x'					
Conv2D†	5×5	2×2	64	LReLU	$\times$
Conv2D†	5×5	2×2	128	LReLU	$\times$
Dense			1	None	$\times$
$\boldsymbol{D_{zz}(z, z')}$					
Concatenate z and z'					
Dense†			64	LReLU	$\times$
Dense†			32	LReLU	$\times$
Dense†			1	LReLU	$\times$
Optimizer	Adam($\alpha = 2 * 10^{-4}, \beta_1 = 0.5$)				
Batch size	32				
Latent dimension	100				
Max Epochs	100				
Patience	10				
LReLU slope	0.2				
Weight & bias initialization	Isotropic gaussian ($\mu = 0, \sigma = 0.01$) Constant(0)				

Table IX

Performance on image data

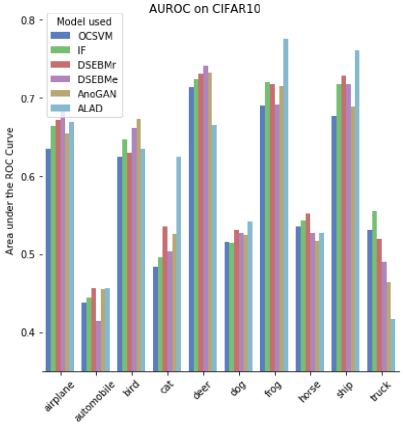


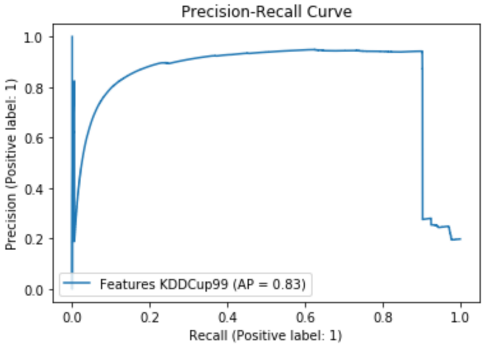
Table of Contents

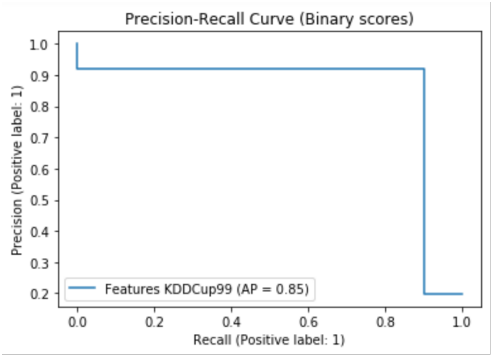
Training setup: KDDCup99

- As in the Paper, due to the high proportion of outliers in the KDD dataset, **"normal"** data samples are treated as **anomalies**.
- **Exactly 19.82 (instead of 20%)** of samples with the highest anomaly scores A(x) are classified as **anomalies** (positive class).
- **Preprocessing**: The dataset contains samples of 41 dimensions, where 34 of them are continuous and 7 are categorical. For categorical features, one-hot representation are used to encode them resulting in a total of 121 features.
- Utilized a **latent discriminator** while the Paper's baseline model not.
- Used metrics: Precision, Recall, F1 score.

Performance on tabular data

Precision: 0.9481 **Recall**: 0.9417 **F1-Score**: 0.9449





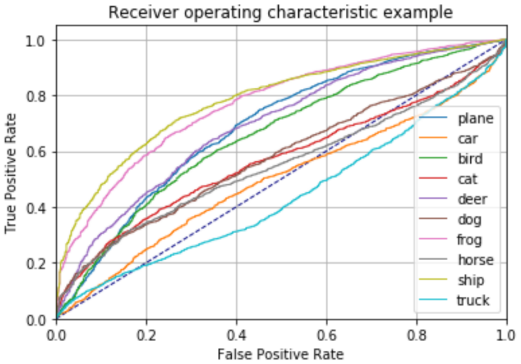
Compared to Paper's result

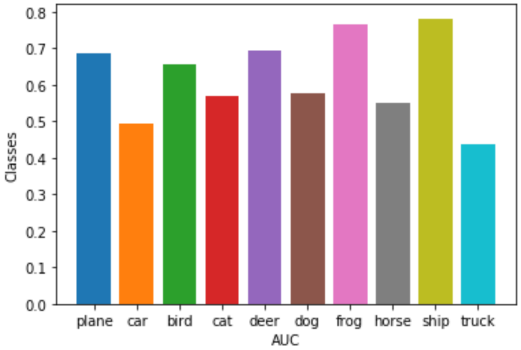
Model	Precision	Recall	F1-Score
OC-SVM	0.7457	0.8523	0.7954
IF	0.9216	0.9373	0.9294
DSEBM-r	0.8532	0.6472	0.7328
DSEBM-e	0.8619	0.6446	0.7399
DAGMM	0.9297	0.9442	0.9369
AnoGAN	0.8786	0.8297	0.8865
ALAD	0.9427	**0.9577**	**0.9501**
Reproduction	**0.9481**	0.9417	0.9449

Experiment Setup

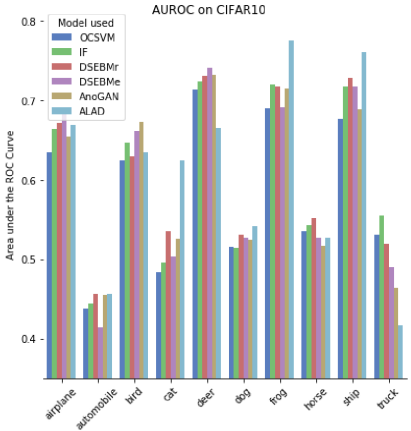
- Ten different datasets from CIFAR-10 are computed where one class is deemed to be **normal** and considering images from the remaining 9 as **anomalous**.
- **Preprocessing:** Pixels were scaled to be in range [-1,1].
- Used metrics: AUROC.
- Utilized a **latent discriminator** while the Paper's baseline model not.

Results


Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
plane
car
bird
cat
deer
dog
frog
horse
ship
truck


Classes
AUC
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
plane
car
bird
cat
deer
dog
frog
horse
ship
truck

Compared to Paper's result



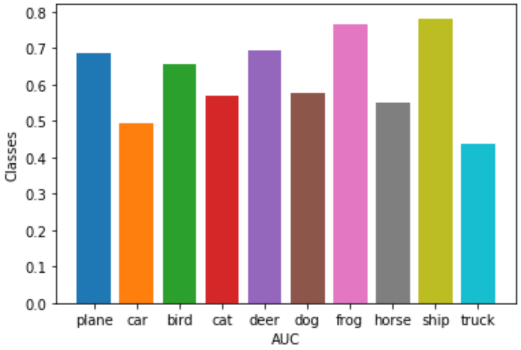


Table of Contents

Take-Home Messages

1. Being able to **learn the distribution of the normal data** is key for the anomaly detection task.
2. **Normal samples** should be **accurately reconstructed** whereas **anomalous samples** will likely be **poorly reconstructed**.
3. Measure distance of original sample and reconstruction and use it as a **anomaly score** to determine whether a sample is anomalous or stems from the normal distribution.

Thank You For Listening!