

# Tutorial on Anomaly Detection

3rd ICFA Beam Dynamics Mini-Workshop on Machine Learning Applications for Particle Accelerators

Antonin Sulc

Hamburg, November 1, 2022

**CDCS**

CENTER FOR DATA AND COMPUTING  
IN NATURAL SCIENCES



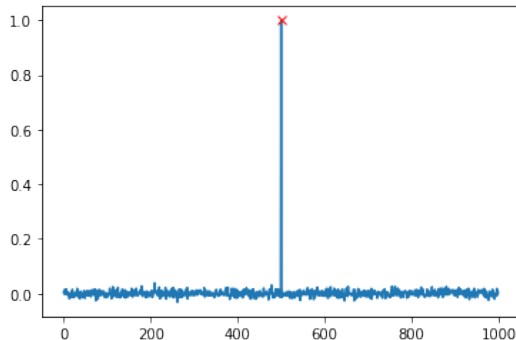
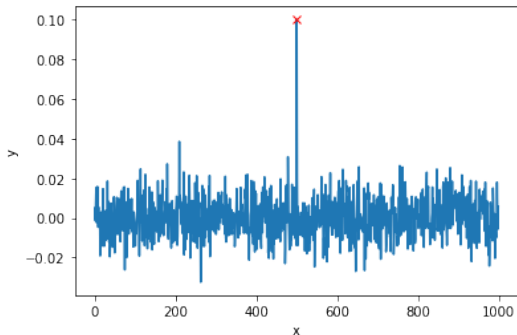
**TUHH**  
*Technische Universität Hamburg*



Universität Hamburg  
DER FORSCHUNG | DER LEHRE | DER BILDUNG

# Point Anomaly

a single data sample that can be considered anomalous compared to the rest of the data [Wittkopp et al.(2022)Wittkopp, Wiesner, Scheinert, and Kao]



# Point Anomaly - Cont'd



EUROPE ECONOMY

## Euro zone inflation hits another record of 9.1% as food and energy prices soar

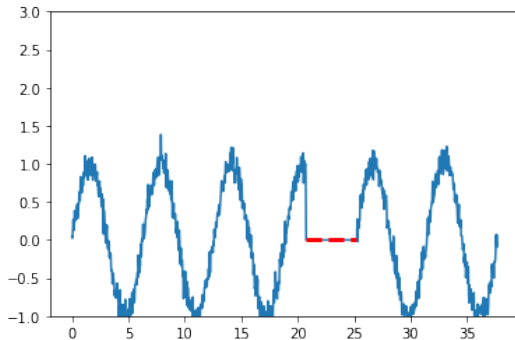
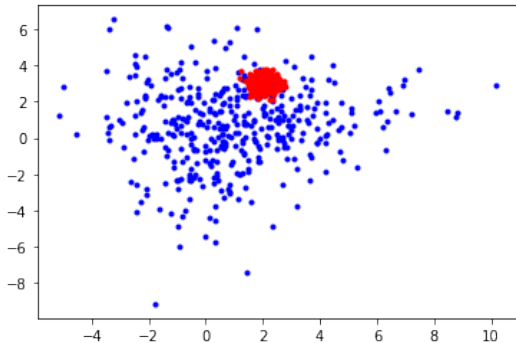
... Eurozone inflation hit a new record high in August of 9.1%, according to flash figures from Europe's statistics office Eurostat, with high energy prices the main driving force. **Werknghwrjhb werbn re4rhn**. The rate was above expectations, with a Reuters poll of economists anticipating a rate of 9%... [Ward-Glenton(2022)]



# Group Anomaly

a multiple data samples that can be considered anomalous compared to the rest of the data

- > Individual samples are often not interesting,
- > An anomalous group of signals is noticeably dense with respect to the entire dataset.



# Contextual Anomaly

samples that are anomalous in a specific context only (but not otherwise) are called **contextual anomalies** [Wittkopp et al.(2022)Wittkopp, Wiesner, Scheinert, and Kao]

- > can have the same feature set (behavioral properties) as normal samples,
- > but are still anomalous within a specific context defined by their contextual properties.

In the context of meadows and suns



Something is wrong with the sun and grass looks artificial.

In the context of Teletubbies



Sun still contains "a person" but it isn't a baby.

# Contextual Anomaly - Cont'd



EUROPE ECONOMY

## Euro zone inflation hits another record of 9.1% as food and energy prices soar

... Eurozone inflation hit a new record high in August of 9.1%, according to flash figures from Europe's statistics office Eurostat, with high energy prices the main driving force. **Pineapples are getting more expensive. Italians can't put them on pizza.** The rate was above expectations, with a Reuters poll of economists anticipating ...

- 1 Talking about pineapple prices isn't relevant although it is money-related.
- 2 Pineapple does not belong to pizza!

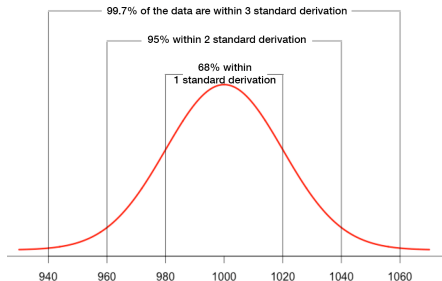


Roll your sleeves!

<https://github.com/sulcantonin/ICFA-Beam-2022>

# Anomaly Detection - Basics - 3-Sigma Rule

- > Mean  $\mu$  defines an average value ( $\mu = 1000$ ).
- > Standard deviation  $\sigma$ , defines how far the normal distribution is spread around the mean ( $\sigma = 20$ ).
- > 68% of all values fall between  $[\mu - \sigma, \mu + \sigma]$ , i.e. [980, 1020].
- > 95% of all values fall between  $[\mu - 2\sigma, \mu + 2\sigma]$ , i.e. [960, 1040].
- > 99.7% of all values fall between  $[\mu - 3\sigma, \mu + 3\sigma]$



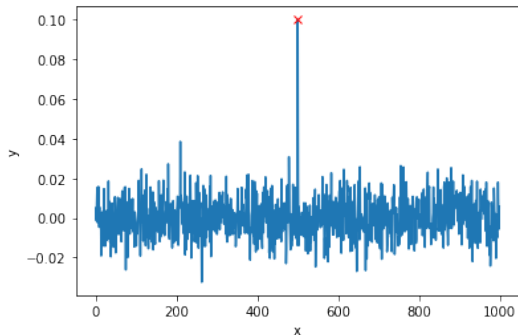
[Magakian(2018)]

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/basics.ipynb>

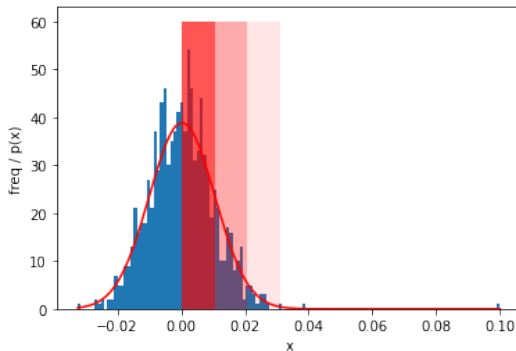


# Anomaly Detection - 3-Sigma Example

Plot



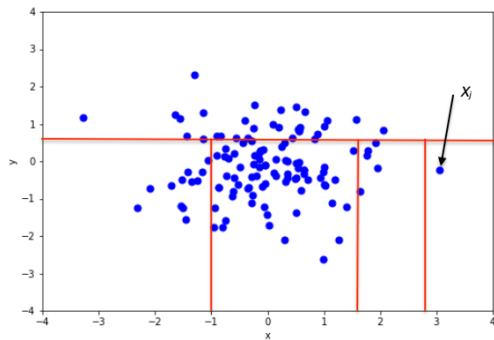
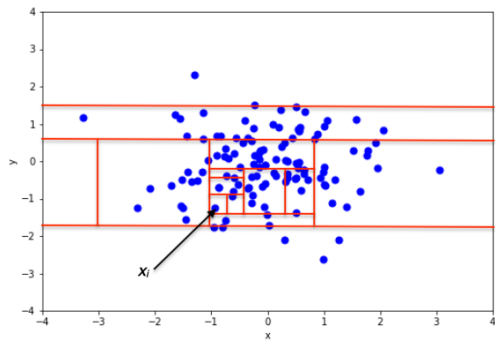
Histogram



See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/basics.ipynb>

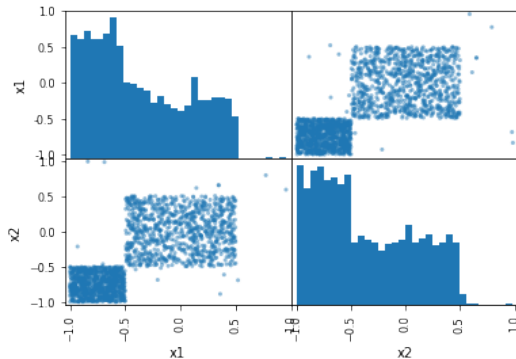
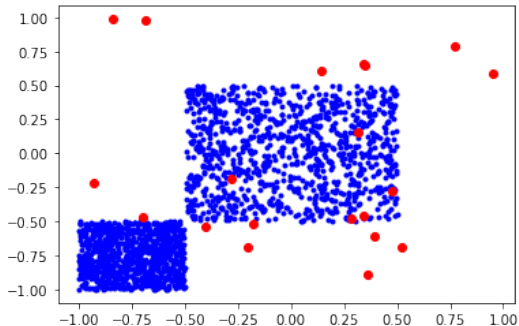
# Anomaly Detection - Isolation Forest

Isolation forest splits the data space using lines that are orthogonal to the origin and assigns higher anomaly scores to data points that need few splits to be isolated. [wik(2022)]

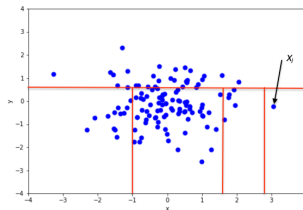
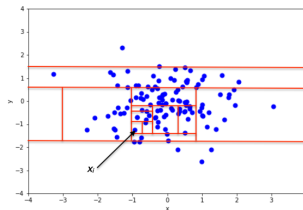
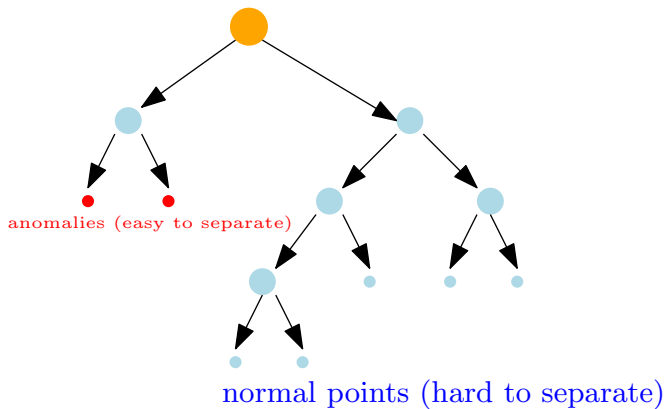


# Anomaly Detection - Isolation Forest

What if the points are not normally distributed?



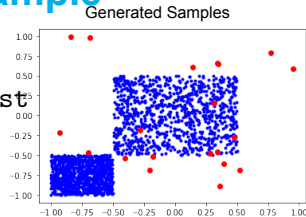
# Anomaly Detection - Isolation Forest



# Anomaly Detection - Isolation Forest Example

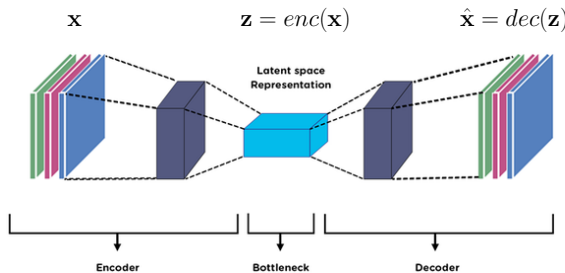
```
from sklearn.ensemble import IsolationForest
import numpy.random as r
r.seed(42)
Xn1 = r.rand(1000,2) - 0.5
Xn2 = (r.rand(1000,2) - 2) * 0.5
Xn = np.concatenate((Xn1,Xn2))
Xa = 2 * (r.rand(20,2) - 0.5)
X = np.concatenate((Xn,Xa))

cont= 20.0 / 2020.0 # ratio of anomalies
f = IsolationForest(contamination = cont, random_state = 42)
l = f.fit_predict(X)
```



# Anomaly Detection - Auto-encoder

What if we have data and want to train a model of a set of data points  $\{x_1 \dots x_N\}$ ?



[Birla(2019)]

$$s = \|\mathbf{x} - \text{dec}(\text{enc}(\mathbf{x}))\|$$

+/- Trains the network "generatively".

- Technically you are not training anomaly detection, but training a model of data

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/autoencoder.ipynb>

## Anomaly Detection - Auto-encoder Example

```
enc = nn.Sequential(  
    nn.Linear(2, 8),  
    nn.ReLU(),  
    nn.Linear(8, 16),  
    nn.ReLU(),  
    nn.Linear(16, 1))  
dec = nn.Sequential(  
    nn.Linear(1, 8),  
    nn.ReLU(),  
    nn.Linear(8, 16),  
    nn.ReLU(),  
    nn.Linear(16, 2))  
  
Reconstr. loss - score  
↓  $\|\mathbf{x} - dec(enc(\mathbf{x}))\|$   
  
def score_reconstruction(x):  
    return vector_norm(dec(enc(x)) - x, dim=-1)
```

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/autoencoder.ipynb>

## Anomaly Detection - Auto-encoder Example Cont'd

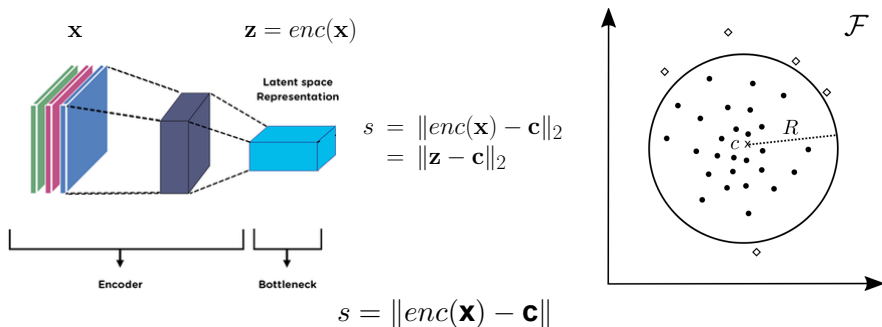
```
optimizer = optim.Adam( list(enc.parameters()) +  
                        list(dec.parameters()))  
  
# reconstruction (X_hat)  
X_hat = model(X)  
# criterion ||X - X_hat||_2  
loss = F.mse_loss(X_hat, X)  
optimizer.zero_grad()  
loss.backward()  
optimizer.step()
```

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/autoencoder.ipynb>



# Anomaly Detection - One Class Loss (OCL)

What if we have data and want to train a model of a set of data points  $\{x_1 \dots x_N\}$ ?



Networks trains to project  $\mathbf{x}$  to fit the hypersphere center  $\mathbf{c}$

[Ruff(2019)]

- + Trains the network "discriminatively",
- One has to be careful with trivial solutions
  - i. e.  $\mathbf{c} \neq 0$ , no biases and unbounded non-linearity

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/oneclass.ipynb>

## Anomaly Detection - One Class Loss

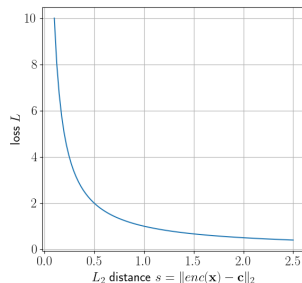
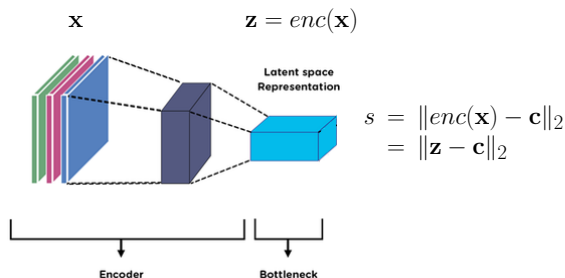
```
enc = nn.Sequential(nn.Linear(2,8, bias = False),
                    nn.Linear(8,16, bias = False),
                    nn.Linear(16,4, bias = False))
# random center of 4D hypersphere
c = torch.randn((1,4), requires_grad = False)
optimizer = optim.Adam(enc.parameters())
# reconstruction (X_hat)
X_hat = enc(X)
loss = vector_norm(X_hat - c, dim = -1).mean()
# Optimisation enc s.t. min ||c - enc(X_hat)||_2
optimizer.zero_grad()
loss.backward()
optimizer.step()
```

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/oneclass.ipynb>

# Anomaly Detection - Semi-Supervised Anomaly Loss (SAL)

What if we have data and want to train a model of a set of data points  $\{x_1 \dots x_N\}$  with very few **known** anomalies?

$$\text{Anomaly} \rightarrow \|enc(\mathbf{x}) - \mathbf{c}\|^{-1}$$



- > Known anomalies ( $l = -1$ ) encourage  $enc$  and  $\mathbf{c}$  to move away.
- > BCE loss minimizes entropy, SAL minimizes entropy of normal samples, and maximizes the entropy of anomalies.

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/sal.ipynb>

## Anomaly Detection - Semi-Supervised Example

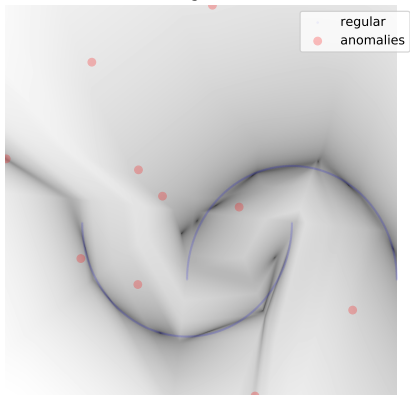
```
enc = nn.Sequential(nn.Linear(2,8),
                    nn.Linear(8,16),
                    nn.Linear(16,4))
# random center of 4D hypersphere
enc.c = nn.Parameter(torch.randn((1,4), requires_grad = True))
optimizer = optim.Adam(enc.parameters())
# reconstruction (X_hat)
X_hat = enc(X)
loss = (vector_norm((X_hat - c), dim = -1)**1).mean()
# Optimisation enc s.t. min ||c - enc(X_hat)||_2
optimizer.zero_grad()
loss.backward()
optimizer.step()
```

See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/sal.ipynb>

# Comparison - Thresholds - Autoencoder + BCE

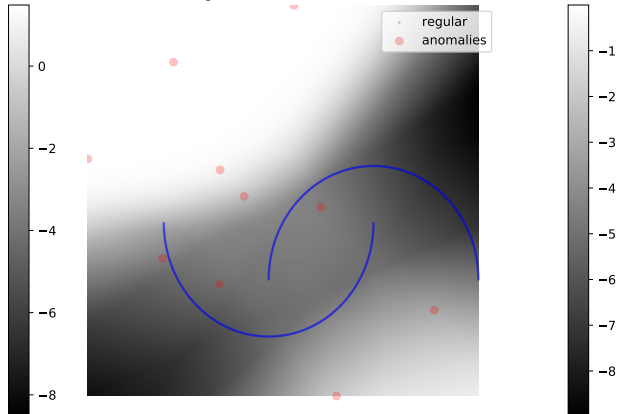
Autoencoder

log score



Binary Cross Entropy

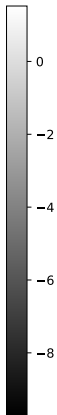
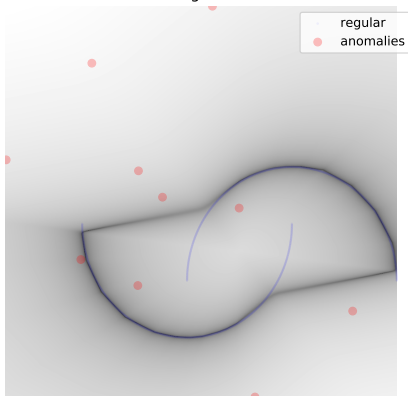
$\log(1 - \text{softmax}(\text{network}(x)))$



# Comparison - Thresholds - OCL + SAL

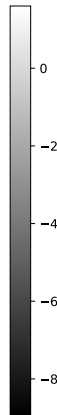
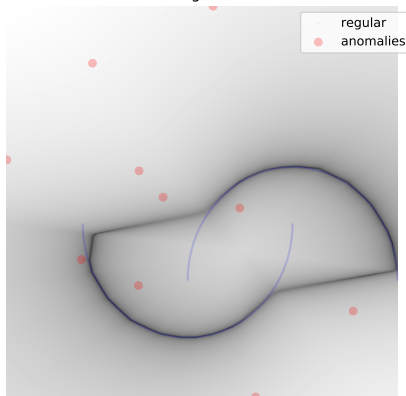
One Class Loss

log score



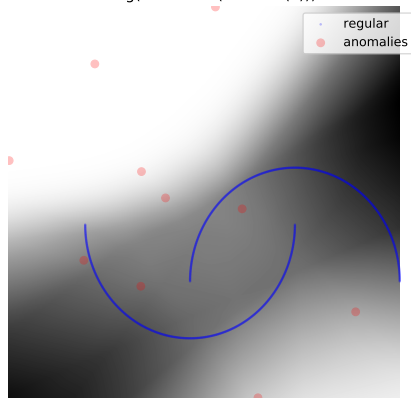
Semi-Supervised

log score

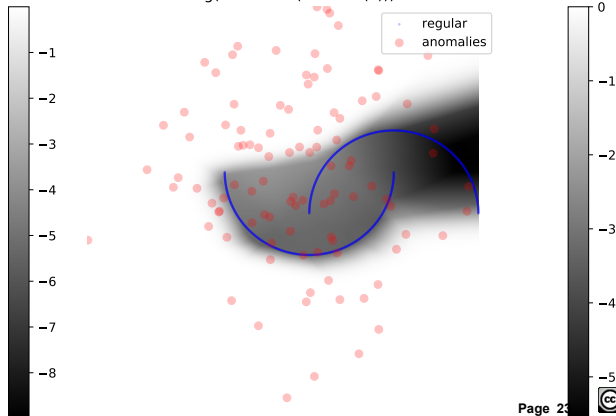


# Comparison - Thresholds - Vanilla Classifier vs Contamination

BCE Contamination 1%  
 $\log(1-\text{softmax}(\text{network}(x)))$

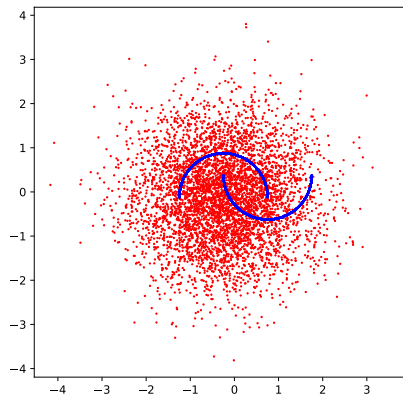


BCE Contamination 10%  
 $\log(1-\text{softmax}(\text{network}(x)))$

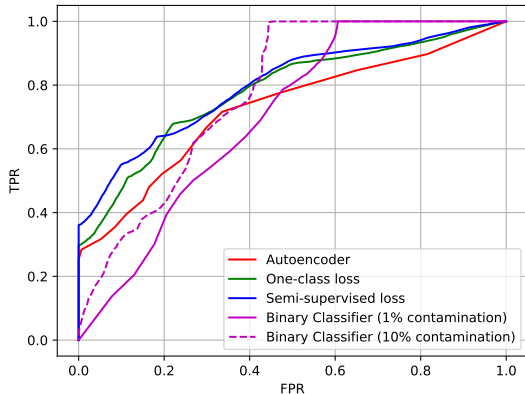


# Comparison - ROC Curve

Test points (50-50% 10k points)

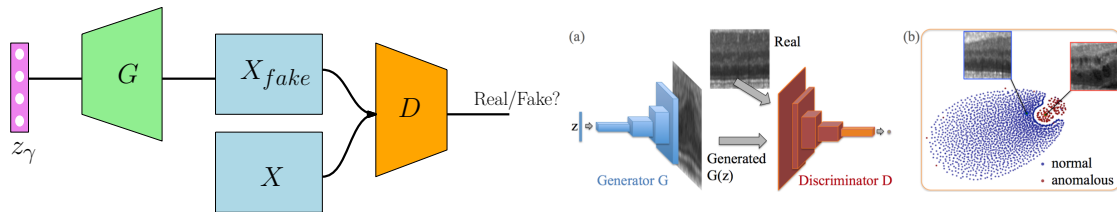


ROC Curves





# Anomaly Detection - Generative Adversarial Networks



- >  $G$  generates **fake** samples and tries to fool discriminator  $D$ .
- >  $D$  receives **fake** and **real samples** and tries to distinguish if an input is fake or real.

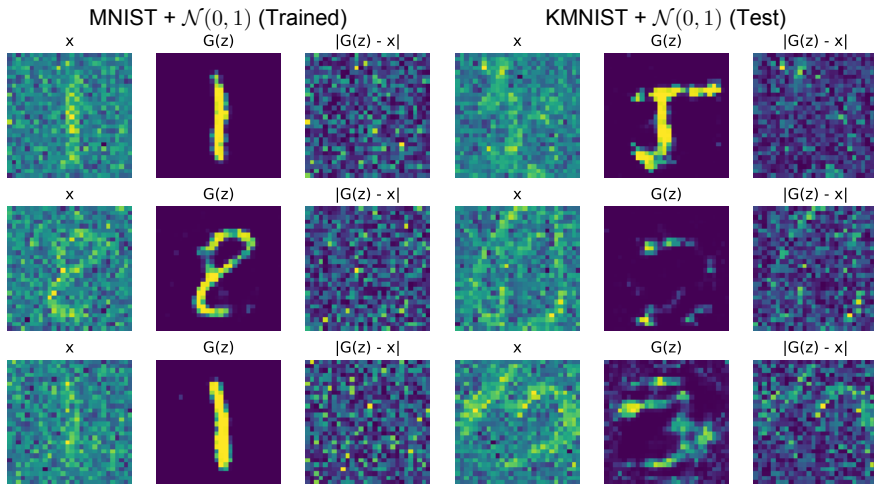
[Goodfellow et al.(2020)Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio, Wiggers(2019),

Schlegl et al.(2017)Schlegl, Seeböck, Waldstein, Schmidt-Erfurth, and Langs]

$$\arg \min_{z_\gamma} \{ (1 - \lambda) \|x - G(z_\gamma)\| + \lambda BCELoss(D(G(z_\gamma)), 1) \}$$

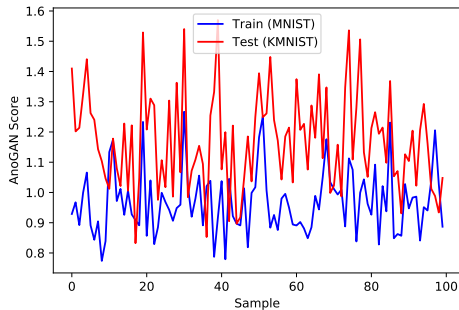
See <https://github.com/sulcantonin/ICFA-Beam-2022/blob/main/gan.ipynb>

# GAN and AnoGAN Example

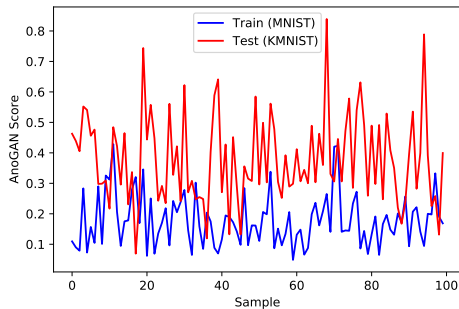


# GAN and AnoGAN Example

## Noisy Inputs Scores



## Noise-Free Samples



# Conclusions


- > 3-Sigma rule if points are normally distributed.
- > Isolation Forests are often also performing well (there are some cases where they work especially well).
- > If you have data where you can create a model you can train a neural network.
- > If you do not have labels use autoencoder or one class loss (OCL).
- > Autoencoder trains model "generatively".
- > If you have labels use semi-supervised one-class loss (SAL) or binary cross entropy loss (BCE).
- > SAL minimizes the entropy of normal samples (non-anomalous, like BCE) but on contrary maximizes the entropy of anomalous samples.

# Thank you!

## Contact

**DESY.** Deutsches  
Elektronen-Synchrotron

[www.desy.de](http://www.desy.de)

Antonin Sulc  
 0000-0001-7767-778X  
MCS DESY  
[antonin.sulc@desy.de](mailto:antonin.sulc@desy.de)

# Bibliography I



Isolation forest, Aug 2022.

URL [https://en.wikipedia.org/wiki/Isolation\\_forest](https://en.wikipedia.org/wiki/Isolation_forest).



Deepak Birla.

Autoencoders, Mar 2019.

URL <https://medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f>.



Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.

Generative adversarial networks.

Communications of the ACM, 63(11):139–144, 2020.

## Bibliography II



Martin Magakian.

Anomaly detection with the normal distribution, Dec 2018.

URL

<https://anomaly.io/anomaly-detection-normal-distribution/index.html>.



L Ruff.

Deep semi-supervised anomaly detection.

arXiv preprint [arXiv:1906.02694](https://arxiv.org/abs/1906.02694), 2019.



Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs.

Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017.

URL <https://arxiv.org/abs/1703.05921>.

## Bibliography III



Hannah Ward-Glenton.

Euro zone inflation hits another record of 9.1% as food and energy prices soar, Aug 2022.

URL [https://www.cnn.com/2022/08/31/](https://www.cnn.com/2022/08/31/euro-zone-inflation-hits-another-record-of-9point1percent-as-food-and-en)

[euro-zone-inflation-hits-another-record-of-9point1percent-as-food-and-en](https://www.cnn.com/2022/08/31/euro-zone-inflation-hits-another-record-of-9point1percent-as-food-and-en)  
[html.](https://www.cnn.com/2022/08/31/euro-zone-inflation-hits-another-record-of-9point1percent-as-food-and-en)



Kyle Wiggers.

Generative adversarial networks: What gans are and how they've evolved, Dec 2019.

URL [https://venturebeat.com/ai/](https://venturebeat.com/ai/gan-generative-adversarial-network-explainer-ai-machine-learning/)

[gan-generative-adversarial-network-explainer-ai-machine-learning/.](https://venturebeat.com/ai/gan-generative-adversarial-network-explainer-ai-machine-learning/)



## Bibliography IV



Thorsten Wittkopp, Philipp Wiesner, Dominik Scheinert, and Odej Kao.  
A taxonomy of anomalies in log data.  
In International Conference on Service-Oriented Computing, pages 153–164.  
Springer, 2022.