Anomaly Detection

Alexey Zaytsev
Assistant professor, Lab Head

Slides by A. Zaytsev & E. Romanenkova

Lecture plan

- Intro to Anomaly detection
- Unsupervised approaches for Anomaly detection. General idea
- Autoencoders for Anomaly detection
- GAN-based Anomaly detection
- Anomaly detection for Time Series

Intro to Anomaly Detection

Problem statement

The problem is to find objects that *anomalous* given training data.

It is ill-specified: small number of labels or no labels.

Normal data

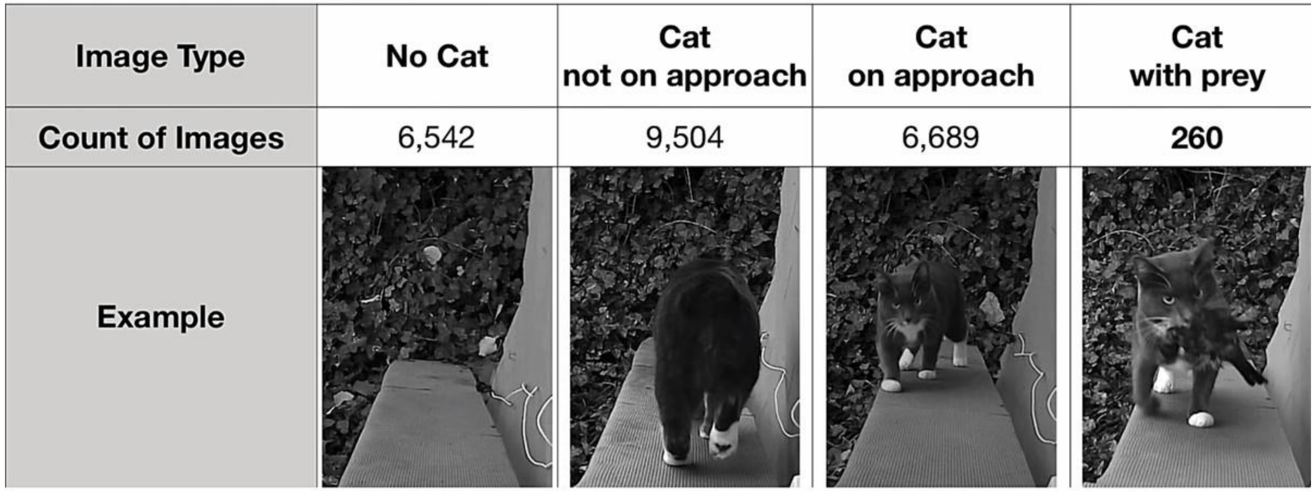
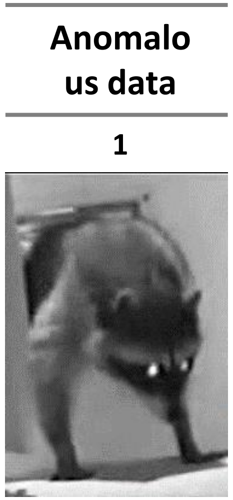


Image Type	No Cat	Cat not on approach	Cat on approach	Cat with prey
Count of Images	6,542	9,504	6,689	**260**
Example				



Anomalous data
1

https://www.theverge.com/tldr/2019/6/30/19102430/amazon-engineer-ai-powered-catflap-prey-ben-hamm

Problem examples are numerous

- Fraud detection 🕵️
- Failure detection for an airplane ✈️
- Intrusion detection 😱
- Earthquake prediction 💥

Problem examples

- Fraud detection 🕵️
- Failure detection for an airplane ✈️
- Intrusion detection 😱
- Earthquake prediction 💥

Typical challenges:

- Requires problem-specific knowledge => new problem – new approach
- Hard to identify something we don’t see
- Bunch of various problem statements => how to define what is anomaly?

Anomaly type taxonomy

Normal data

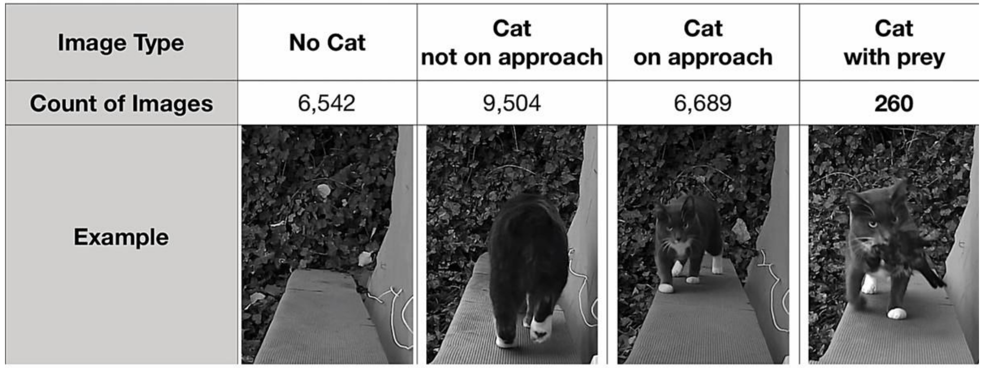
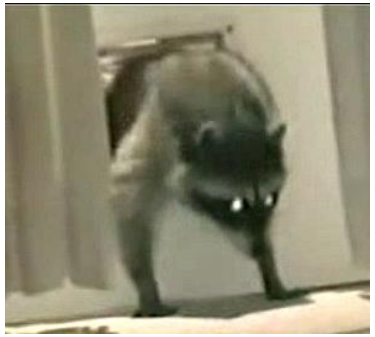
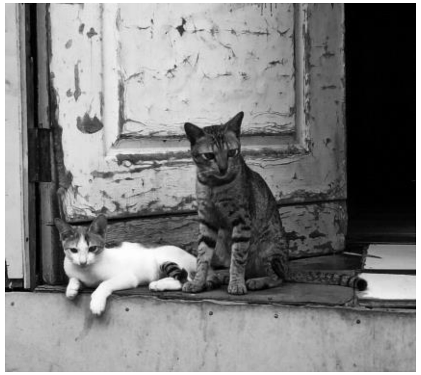


Image Type	No Cat	Cat not on approach	Cat on approach	Cat with prey
Count of Images	6,542	9,504	6,689	**260**
Example				

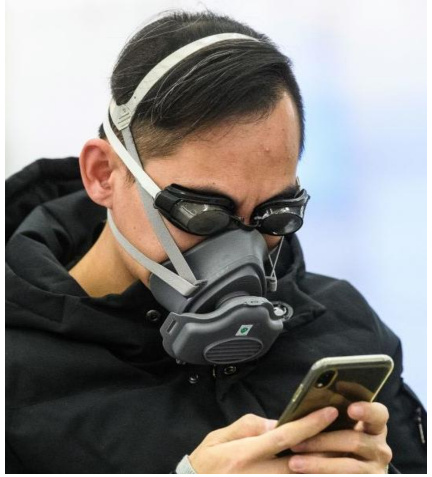
Point Anomaly



Group Anomaly



Contextual Anomaly



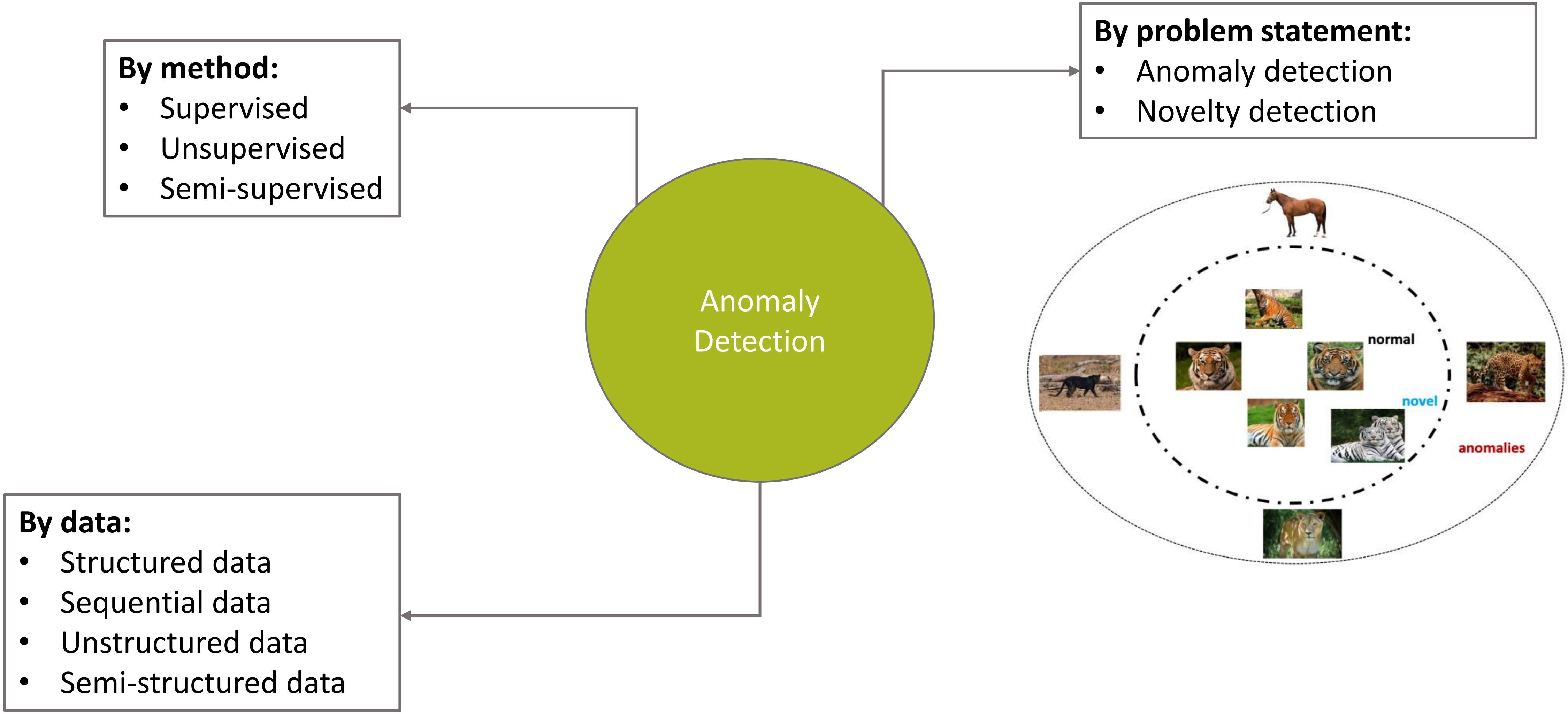
In 2019

He is mad! We should avoid him. (anomaly)

In 2020

He takes care of himself and others. Well done! (normal)

Different taxonomies



Anomaly Detection

By method:
- Supervised
- Unsupervised
- Semi-supervised

By problem statement:
- Anomaly detection
- Novelty detection

By data:
- Structured data
- Sequential data
- Unstructured data
- Semi-structured data

https://arxiv.org/pdf/1901.03407.pdf

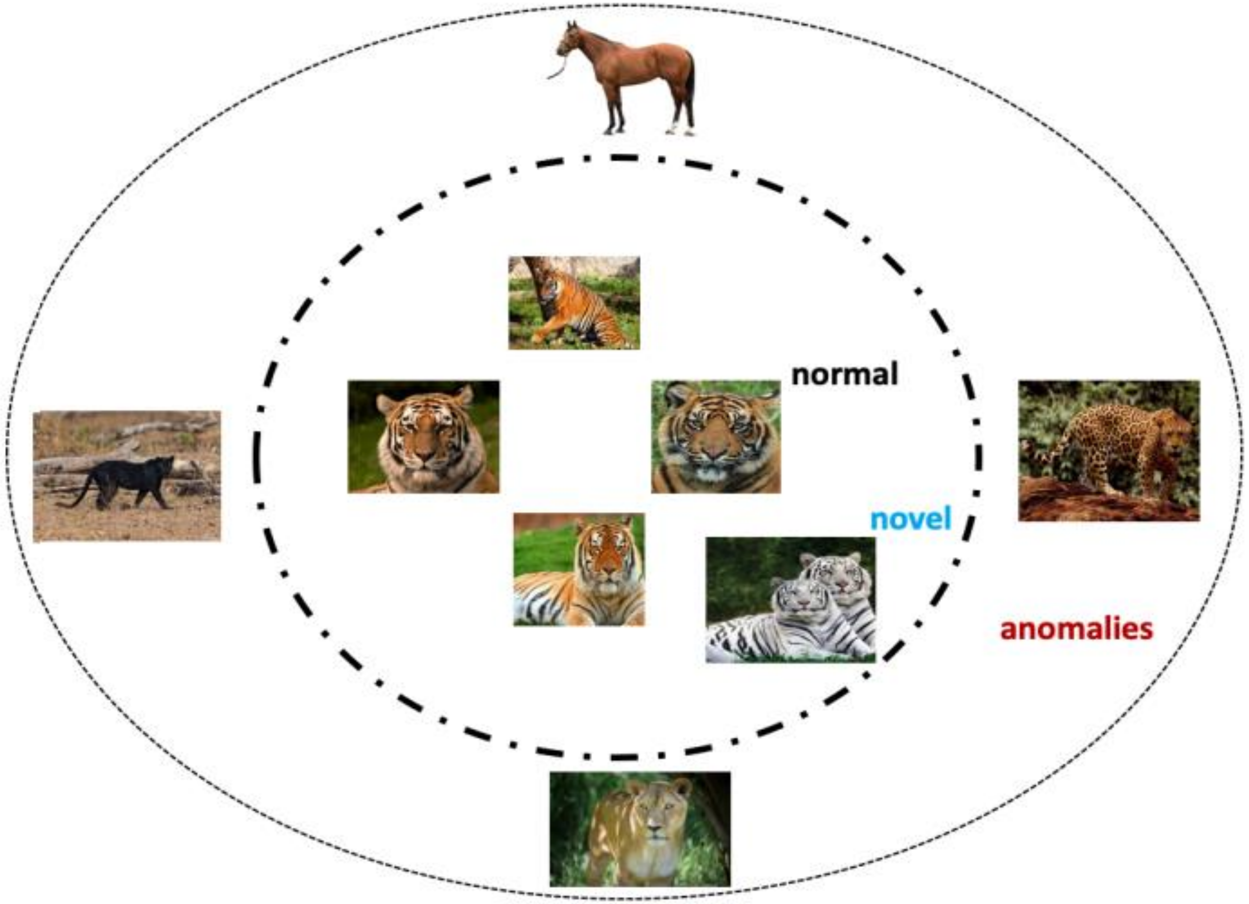
Taxonomy by type of anomalies

Novelty detection

We have never seen the objects of this class

Anomaly detection

Outliers, can be already in the sample



https://arxiv.org/pdf/1901.03407.pdf

Well-defined anomaly assumption – a more formal approach

WDAD [Tom Dietterich]: the anomalies are drawn from a well-defined probability distribution

Example: repeated instances of known machine failures

The WDAD assumption is often risky:

- adversarial situations (fraud, insider threats, cyber security)
- diverse set of potential causes (novel device failure modes)
- user’s notion of “anomaly” changes with time (e.g., anomaly == “interesting point”)

LOOKING UNDER THE LAMPPOST
LOST YOUR KEYS?
YEAH, I LOST THEM OVER THERE BUT THE LIGHT'S BETTER HERE
sketchplanations

https://images.prismic.io/sketchplanations/54100eae-1fee-40b7-b3d7-c0b9e309da1c_SP+711+-+Looking+under+the+lamppost.png?auto=compress,format

https://web.engr.oregonstate.edu/~tgd/talks/dietterich-cmu-2020-anomaly-detection.pdf

Supervised anomaly detection

Supervised Anomaly detection is an imbalanced classification

Weights for classes

- *Proved not to be helpful in most cases*

Resampling methods

- *Undersampling*
- *Oversampling/data generation: SMOTE, etc.*
- *Generative models*

How to choose which method to use?
How to choose resampling parameter?

Specific models
e.g. SMOTEboost, DeepSMOTE

Specific metrics

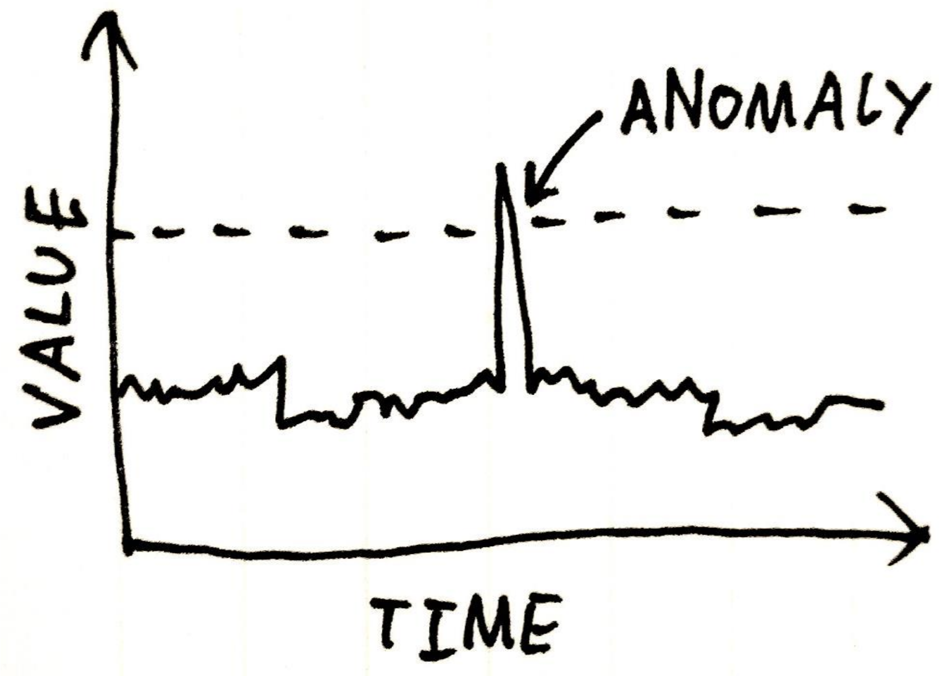
Unsupervised kernel-based Anomaly Detection

Classic pipeline for anomaly detection

1. Construct an anomaly score *s(x)* using data

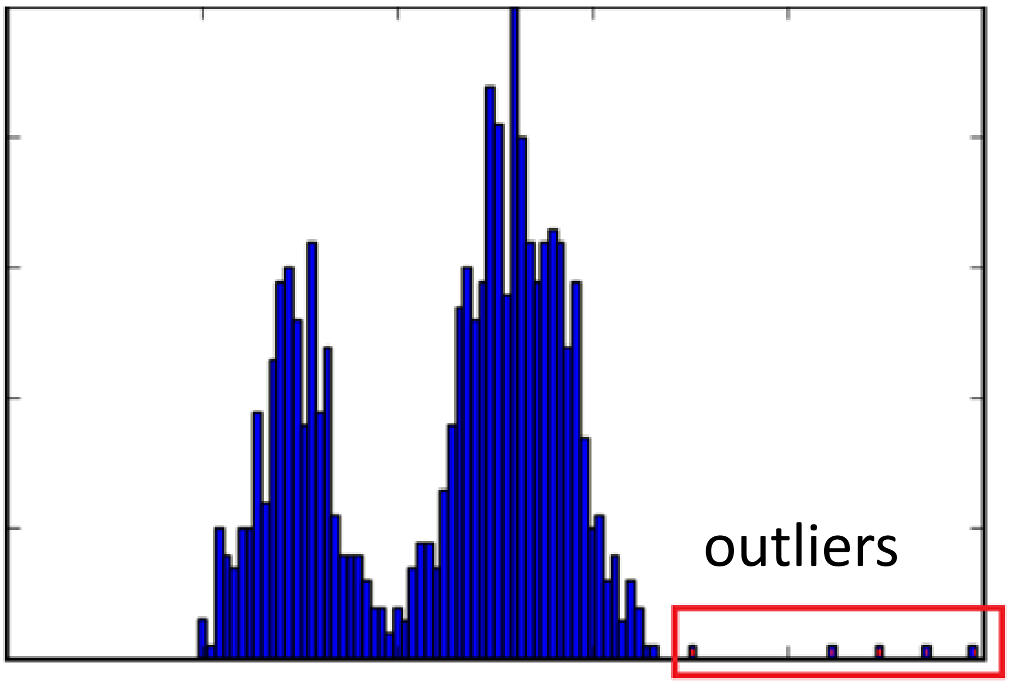
2. Signal about anomaly if the anomaly score is greater than some threshold τ

Threshold selection τ is a separate problem, as we often have only positive examples

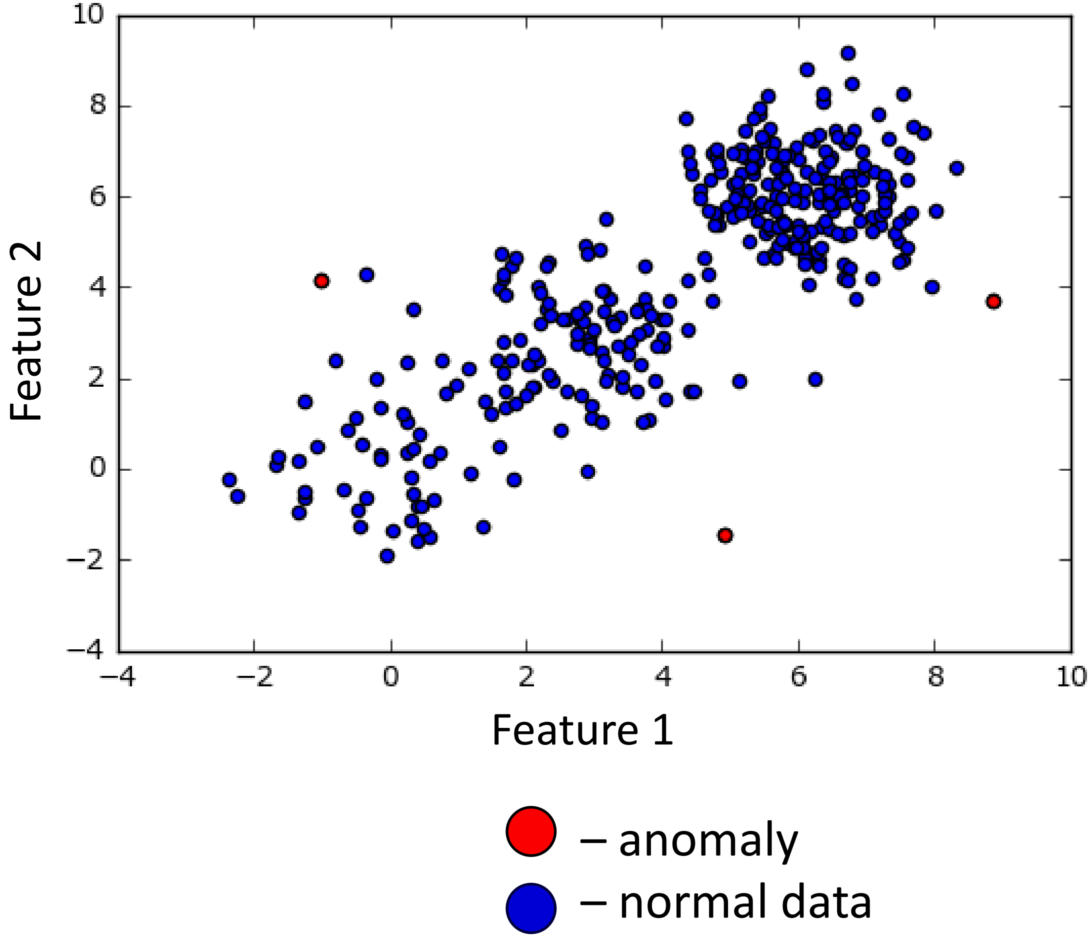


Anomaly detection with a keen eye

One feature anomaly

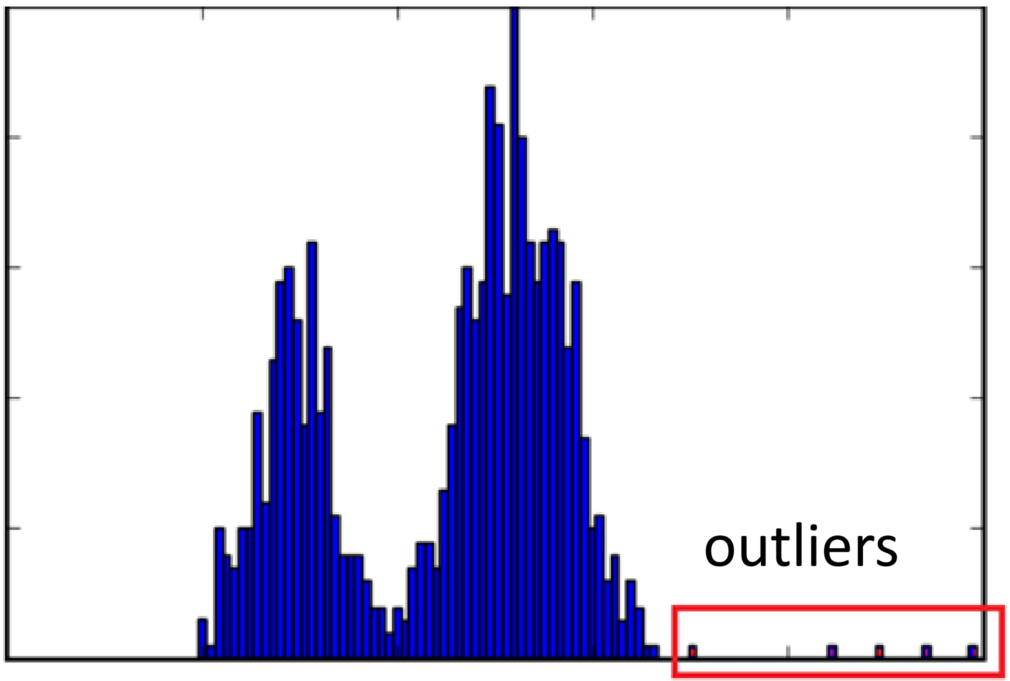


Pair of features anomaly

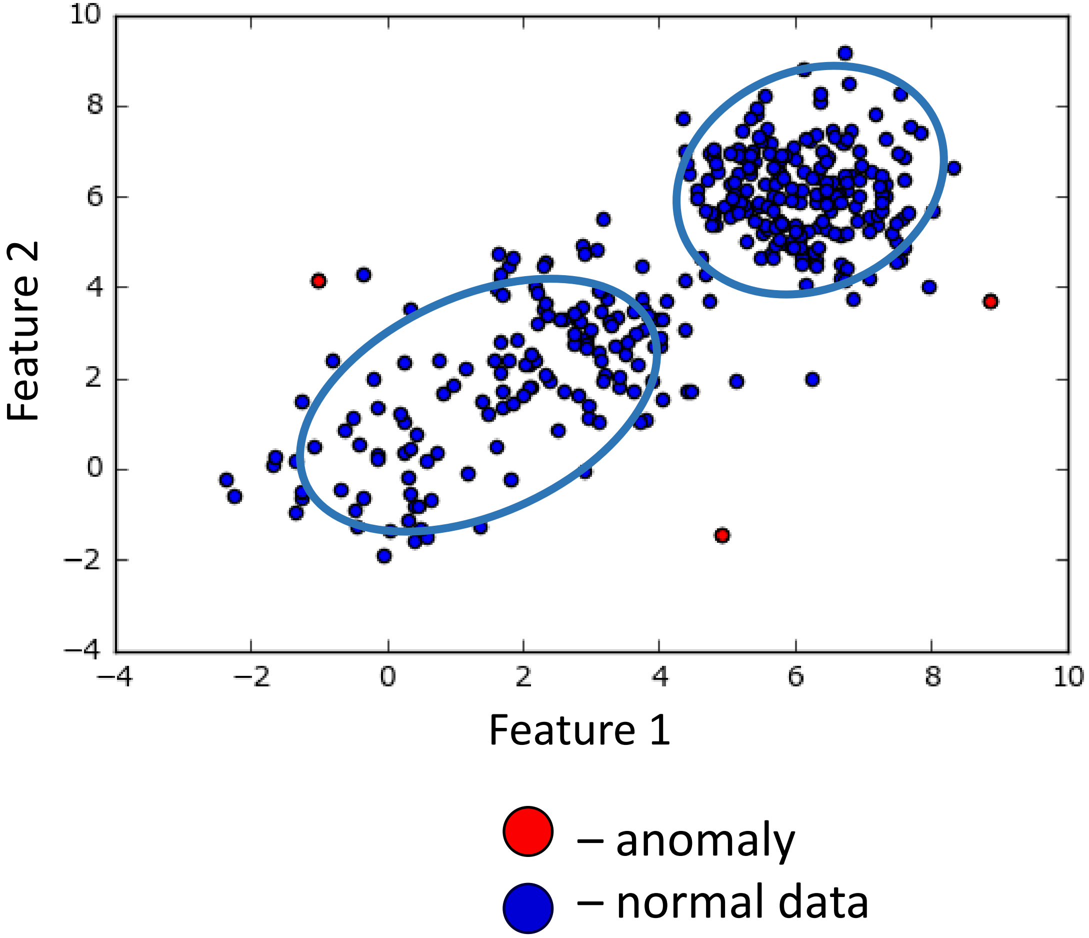


Formal interpretation: low density corresponds to outliers

One feature anomaly



Pair of features anomaly



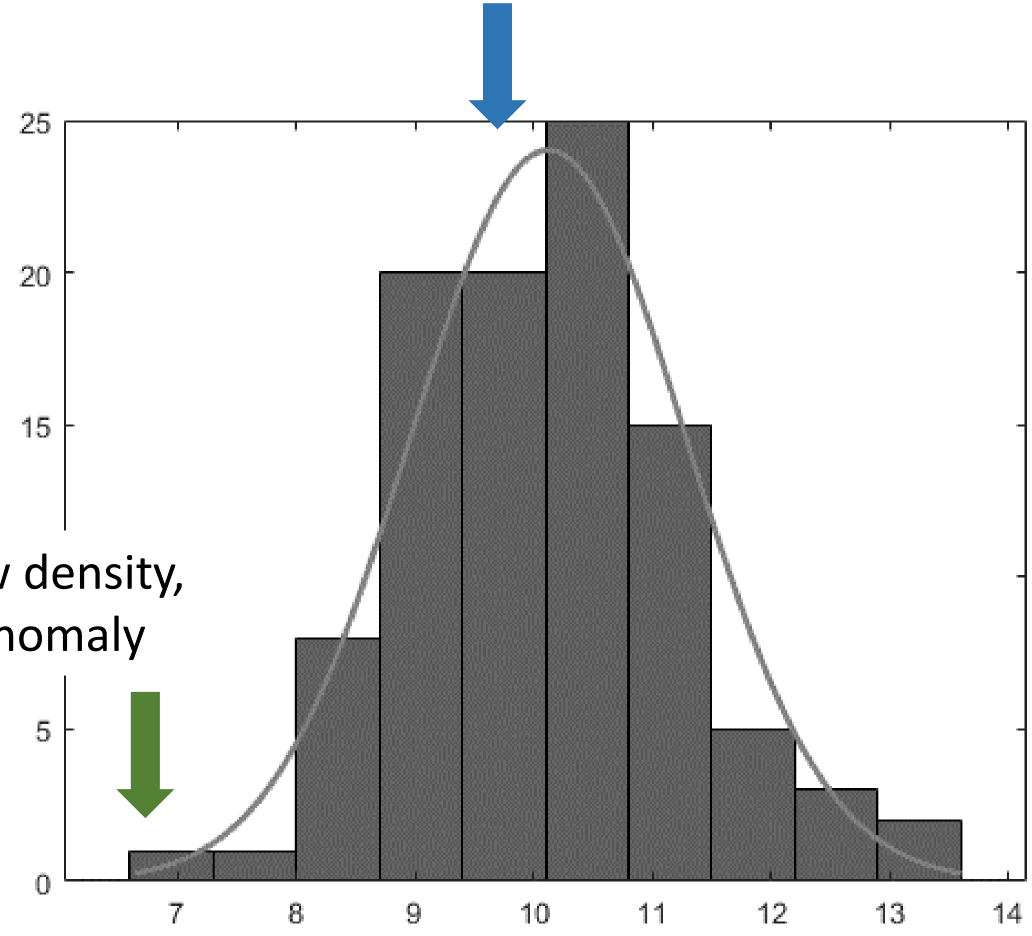
Anomaly detection using a single feature

Histogram: a step-wise density estimation

Parametric approach: the curve is the recovered density for data $\mathrm{p}(x)$

High density, normal

$\mathrm{p}(x)$

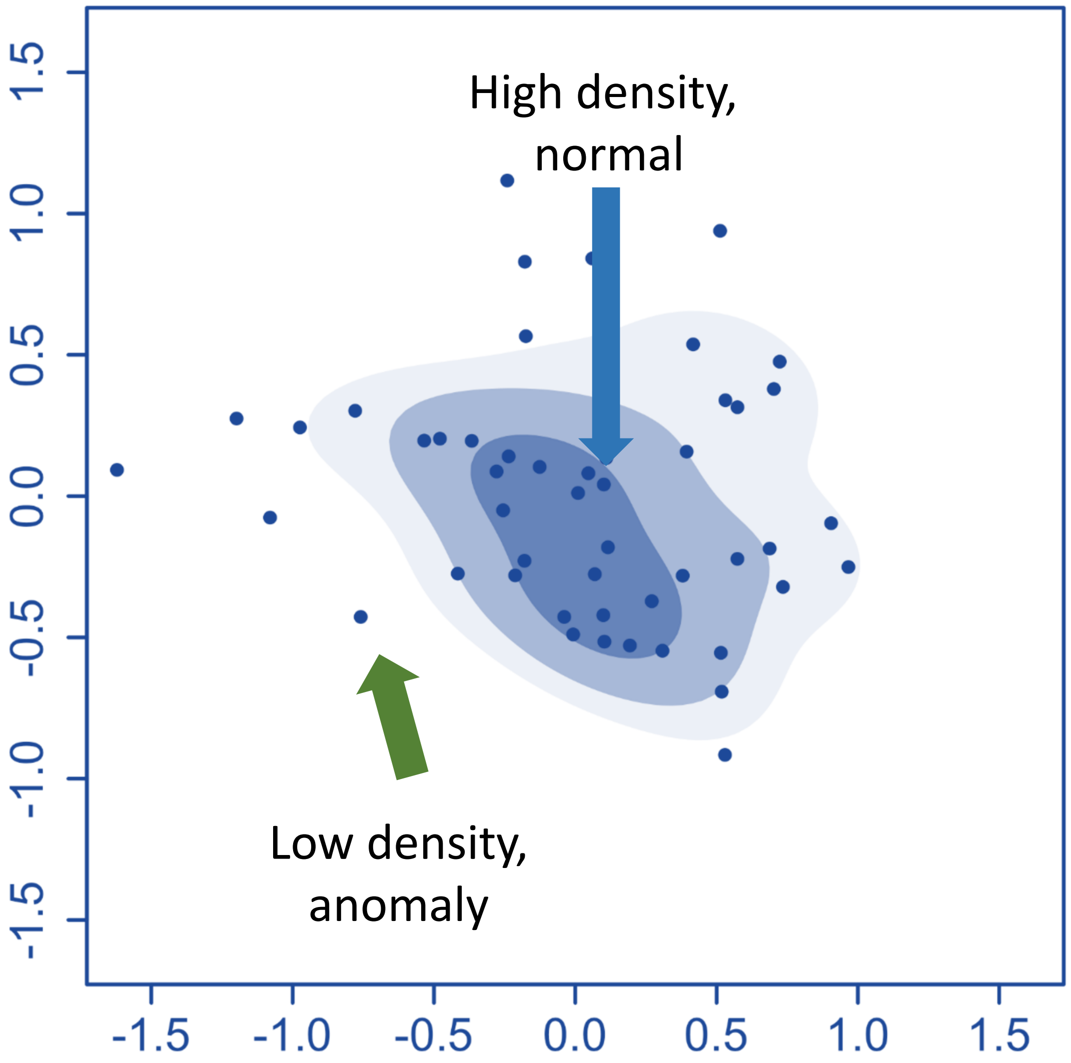


Low density, anomaly

Anomaly detection using a pair of features

Points are real data

Shading corresponds to a density estimation


High density, normal

Low density, anomaly

Kernel density estimation

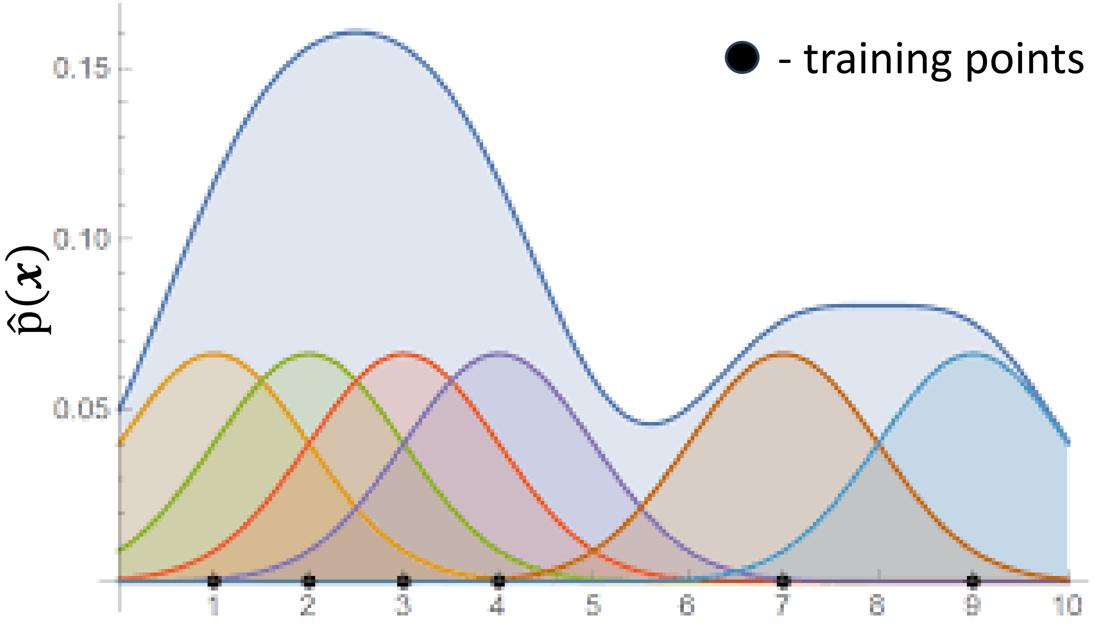
We can estimate the density $p(\boldsymbol{x})$ for a low input dimension easily

A kernel method is a key:

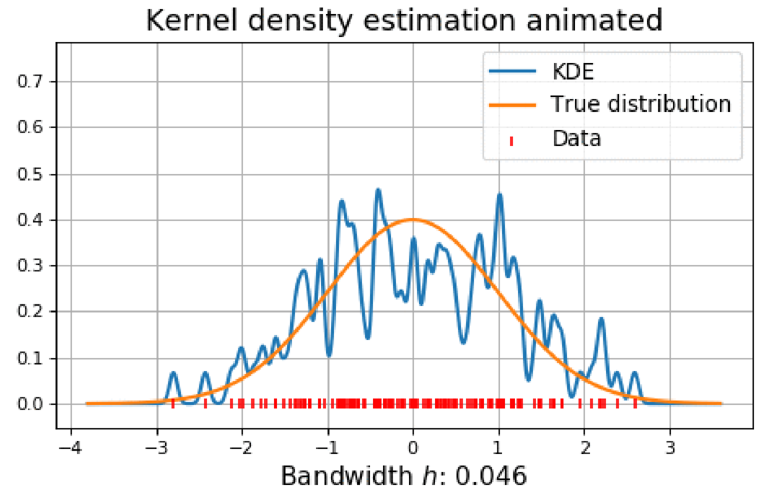
$$\hat{\mathrm{p}}(\boldsymbol{x}) = \frac{1}{n}\sum_{\boldsymbol{x}'} \mathrm{k}(\boldsymbol{x} - \boldsymbol{x}'),$$

$$k(\boldsymbol{x}-\boldsymbol{x}') = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{2\sigma^2}\right)$$

We can vary kernel width σ to maximize the fit quality (but it is out of scope of today’s lecture)



https://ekamperi.github.io/math/2020/12/08/kernel-density-estimation.html



https://github.com/fbeilstein/machine_learning/blob/master/lecture_15_kernel_density_estimation.ipynb

LODA anomaly detector

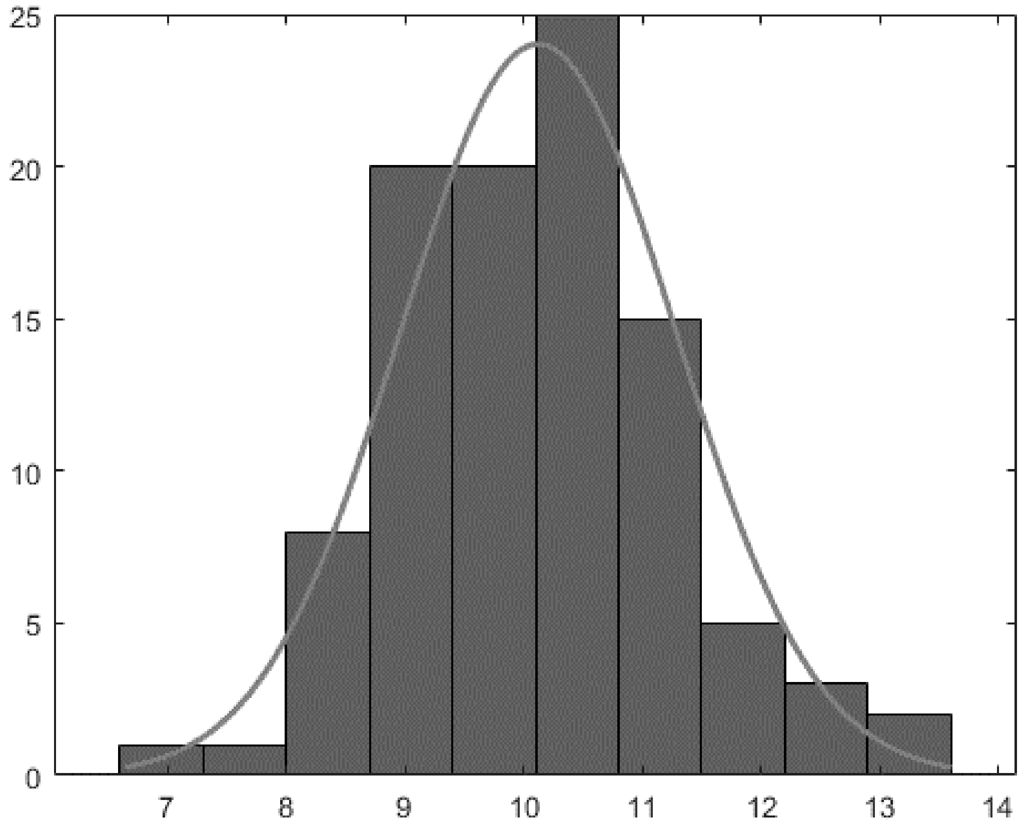
1. Build *M* random linear projections

2. Estimate density for each projection $\mathrm{p}_i(\boldsymbol{x})$

3. Mean density for all projections

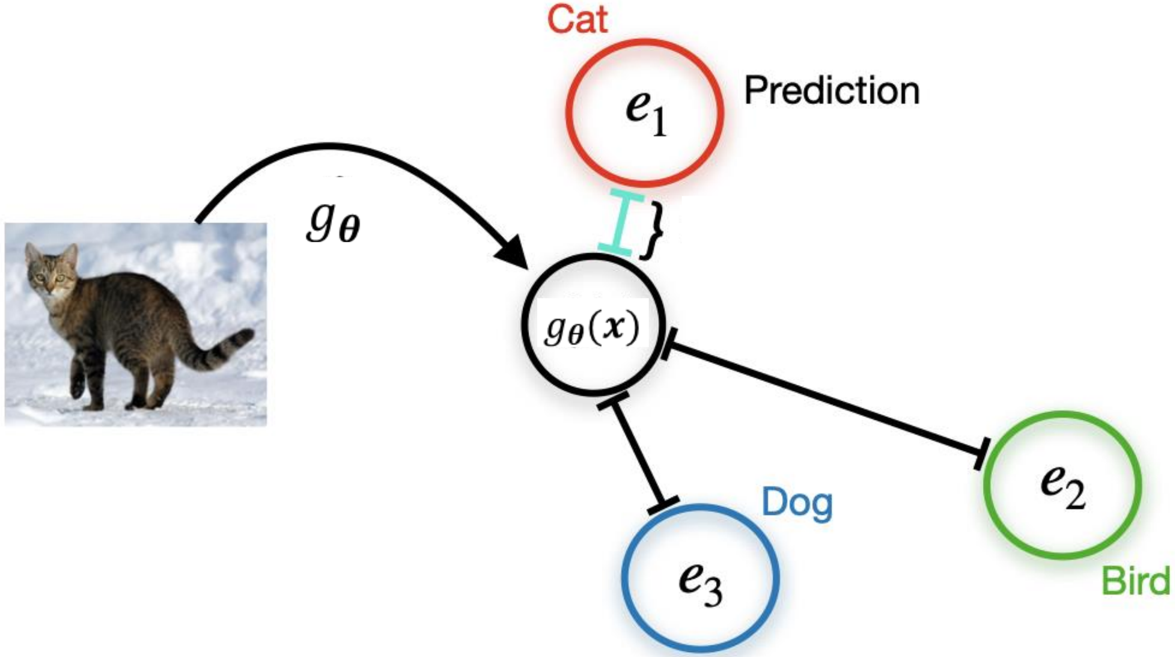
$$\mathrm{S}(x) = -\frac{1}{M}\sum_{i=1} \log \mathrm{p}_i(\boldsymbol{x})$$

Get mean «surprise» $\mathrm{E}\,\mathrm{S}(\boldsymbol{x})$



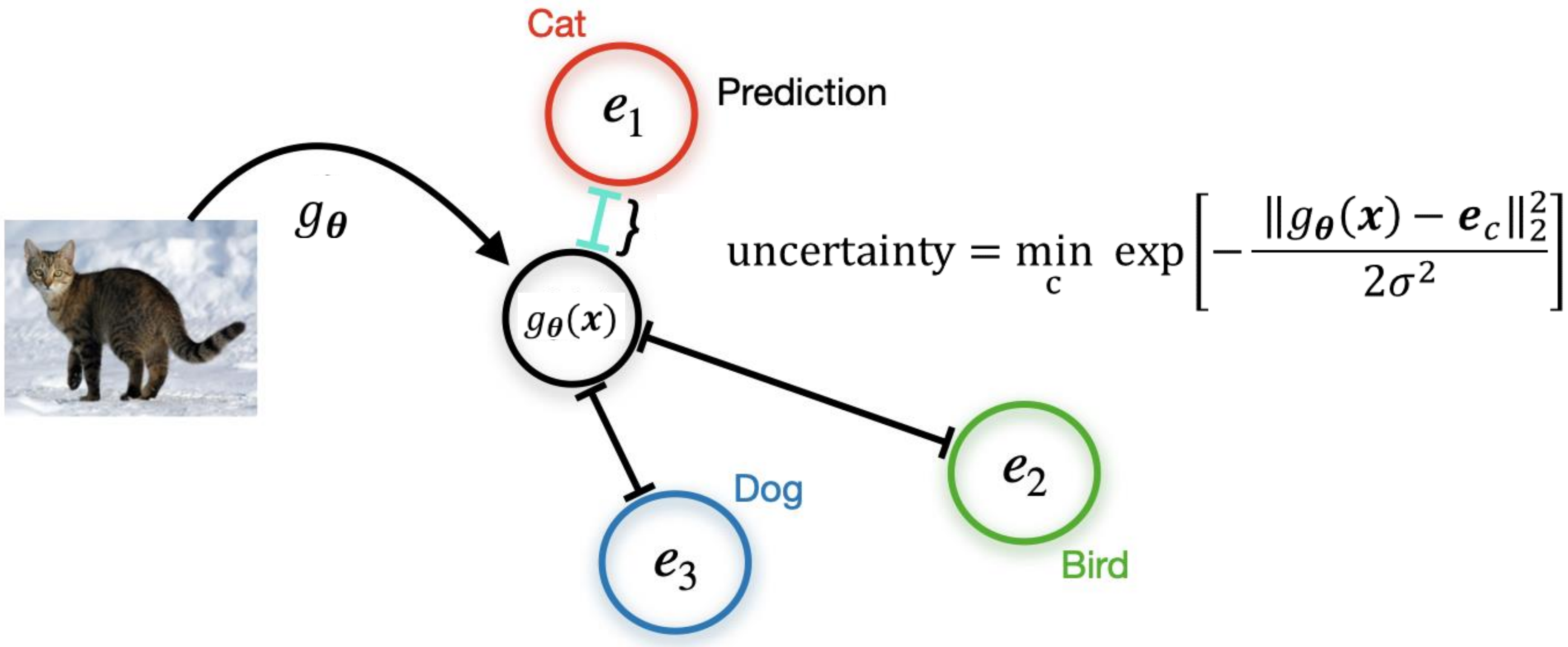
Representations from Neural Networks

- $g_\theta(\boldsymbol{x})$ is a neural network backbone
- $\boldsymbol{e}_i$ is the mean embedding for i-th class



Source: Van Amersfoort, Joost, et al. "Uncertainty estimation using a single deep deterministic neural network." *ICML*, 2020.

Utilize representations for anomaly detection


Cat
e_1
Prediction
$g_{\boldsymbol{\theta}}$
$g_{\boldsymbol{\theta}}(\boldsymbol{x})$
e_2
Bird
e_3
Dog

$$\text{uncertainty} = \min_{c} \; \exp\left[-\frac{\|g_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{e}_c\|_2^2}{2\sigma^2}\right]$$

A DUQ architecture: the input is mapped to the feature space, where it is assigned to the closest centroid. The distance to this centroid is uncertainty.

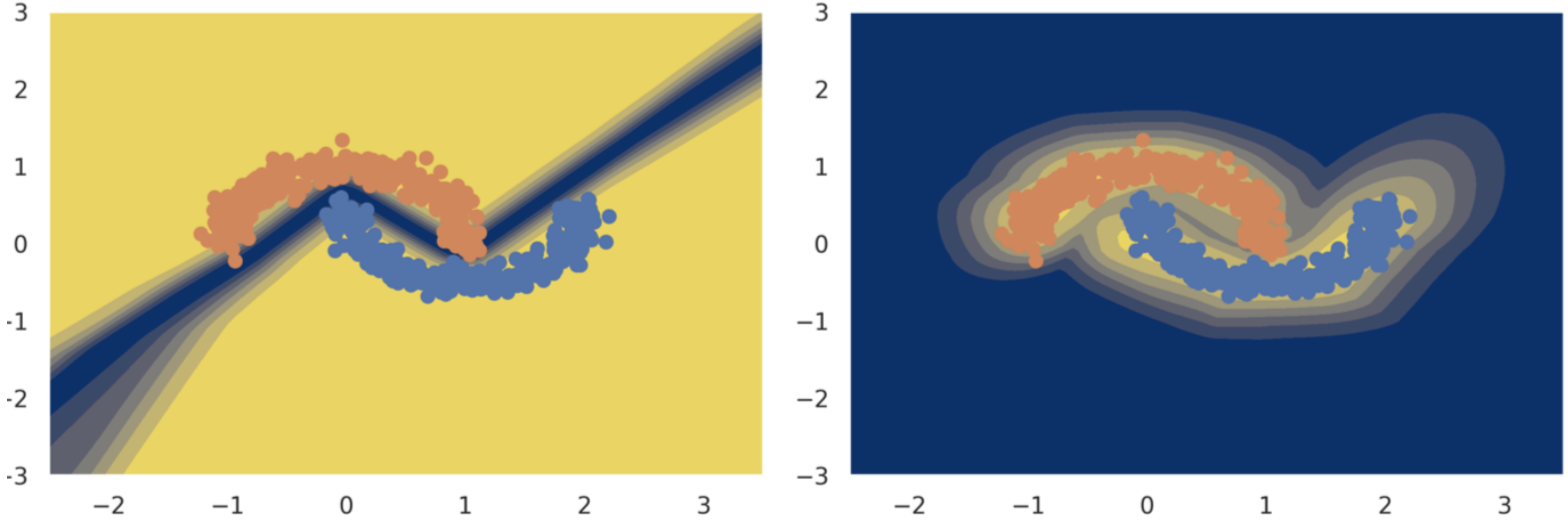
Source: Van Amersfoort, Joost, et al. "Uncertainty estimation using a single deep deterministic neural network." *ICML*, 2020.

Toy example for DUQ

🟦 High anomaly score

🟨 Low anomaly score

🟠 🔵 Training data



Baseline

DUQ model

Source: Van Amersfoort, Joost, et al. "Uncertainty estimation using a single deep deterministic neural network." *ICML*, 2020.

Forest-based Anomaly Detection

Top recommendations from the three studies

- Isolation based methods: iForest and iNNE [8]
- All known weaknesses of iForest are overcome by iNNE, with higher time cost; but still has significantly lower time cost than kNN.
- Nearest neighbour-based methods: aNNE and kNN
- Clustered anomalies: ABOD & LOF
- iNNE [5] can do well in detecting clustered anomalies.
- Kernel Mahalanobis [6]: The key weakness is the time cost

- Among the compared methods, iForest and iNNE have the highest detection accuracy and also have the lowest time cost.

[6] Hoffmann, H. (2007). Kernel PCA for Novelty Detection, Pattern Recognition, 40(3), 863–874.

[8] Bandaragoda, T. R., Ting, K. M., Albrecht, D., Liu F. T., Wells, J. R. (2018). Isolation-based Anomaly Detection using Nearest Neighbour Ensembles. Computational Intelligence. Doi:10.1111/coin.12156.

https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

Models based on ensembles of decision trees

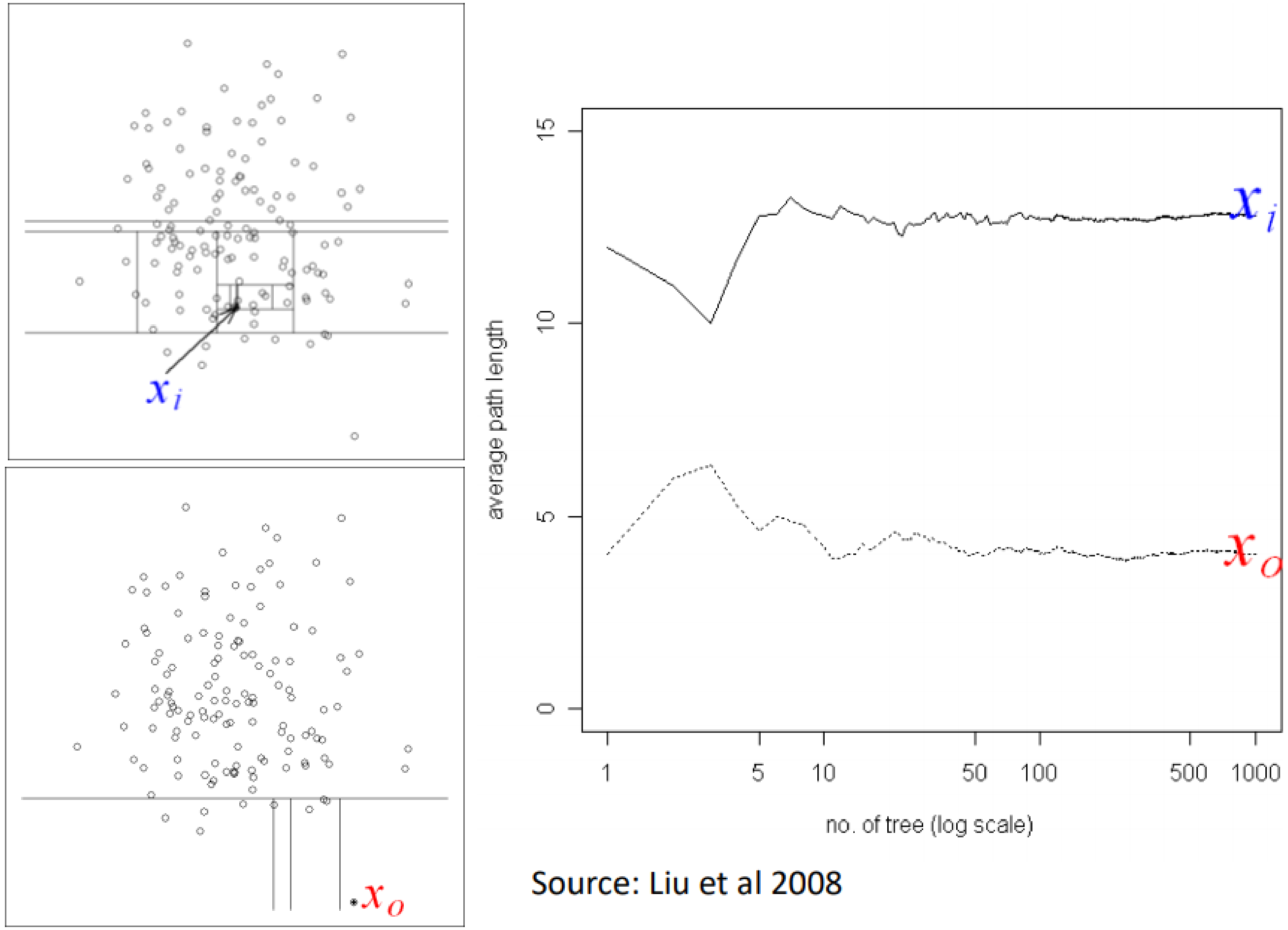
- Isolation forest **iForest**

- Isolation Nearest Neighbours Ensembles **INNE**

https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

Isolation tree

- We draw random split until a node is separated
- We calculate average path length
- It is the normality score



Source: Liu et al 2008

[Liu et al ICDM 2008]

https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

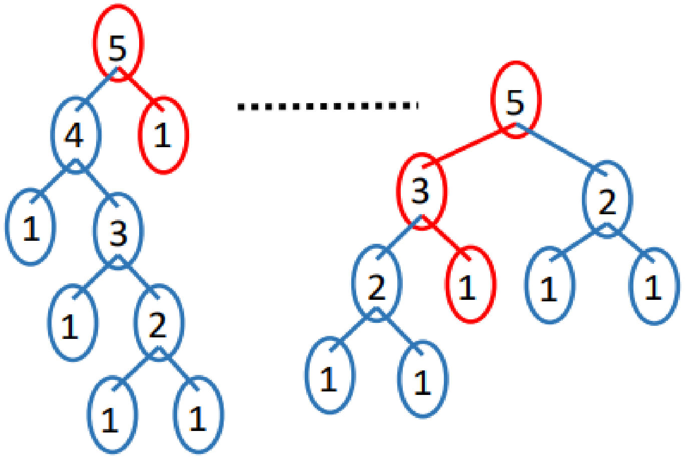
Isolation forest

- A collection of isolation trees (iTrees)

- Each iTree isolates every instance from the rest of the instances in a given sample

- Anomalies are ‘few and different’
  - More susceptible to isolation
  - Shorter average path

$$Score(\boldsymbol{x}) = \frac{1}{t}\sum_{i=1}^{t} \ell_i(\boldsymbol{x})$$

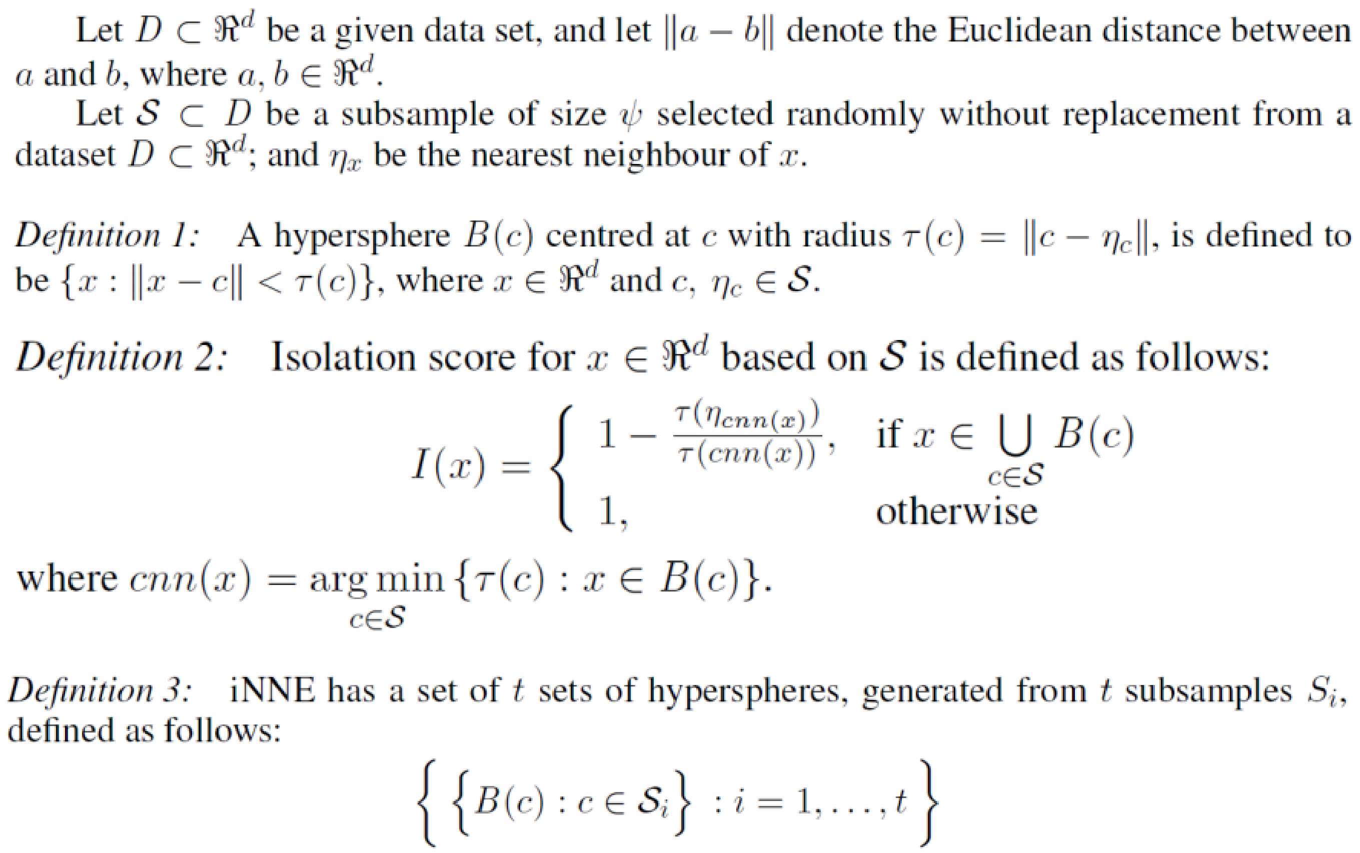


where $\ell_i(\boldsymbol{x})$ is the path length of $\boldsymbol{x}$ traversed in tree i

[Liu et al ICDM 2008]

https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

INNE approach



Let $D \subset \Re^d$ be a given data set, and let $\|a - b\|$ denote the Euclidean distance between a and b, where $a, b \in \Re^d$.

Let $\mathcal{S} \subset D$ be a subsample of size ψ selected randomly without replacement from a dataset $D \subset \Re^d$; and η_x be the nearest neighbour of x.

Definition 1: A hypersphere $B(c)$ centred at c with radius $\tau(c) = \|c - \eta_c\|$, is defined to be $\{x : \|x - c\| < \tau(c)\}$, where $x \in \Re^d$ and $c,\ \eta_c \in \mathcal{S}$.

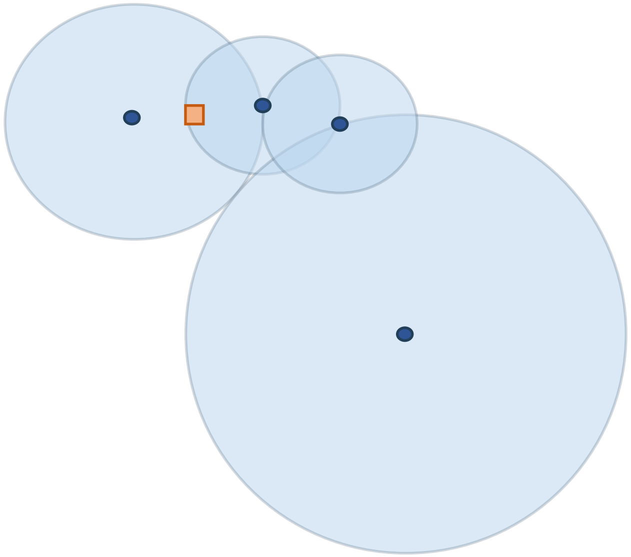
Definition 2: Isolation score for $x \in \Re^d$ based on $\mathcal{S}$ is defined as follows:

$$I(x) = \begin{cases} 1 - \frac{\tau(\eta_{cnn(x)})}{\tau(cnn(x))}, & \text{if } x \in \bigcup_{c \in \mathcal{S}} B(c) \\ 1, & \text{otherwise} \end{cases}$$

where $cnn(x) = \arg\min_{c \in \mathcal{S}} \{\tau(c) : x \in B(c)\}$.

Definition 3: iNNE has a set of t sets of hyperspheres, generated from t subsamples S_i, defined as follows:

$$\left\{ \left\{ B(c) : c \in \mathcal{S}_i \right\} \ : i = 1, \dots, t \right\}$$



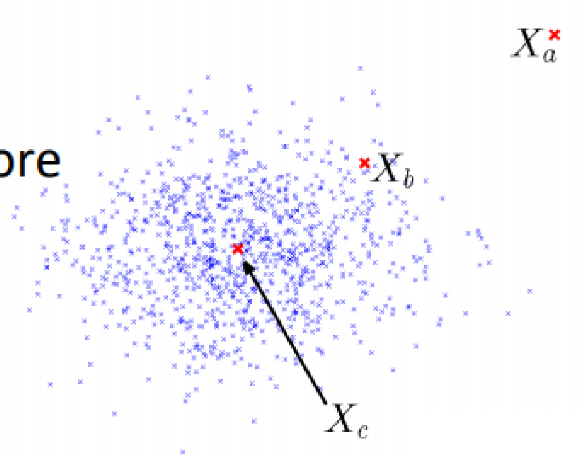
The final anomaly score is the mean value of subsample anomaly scores

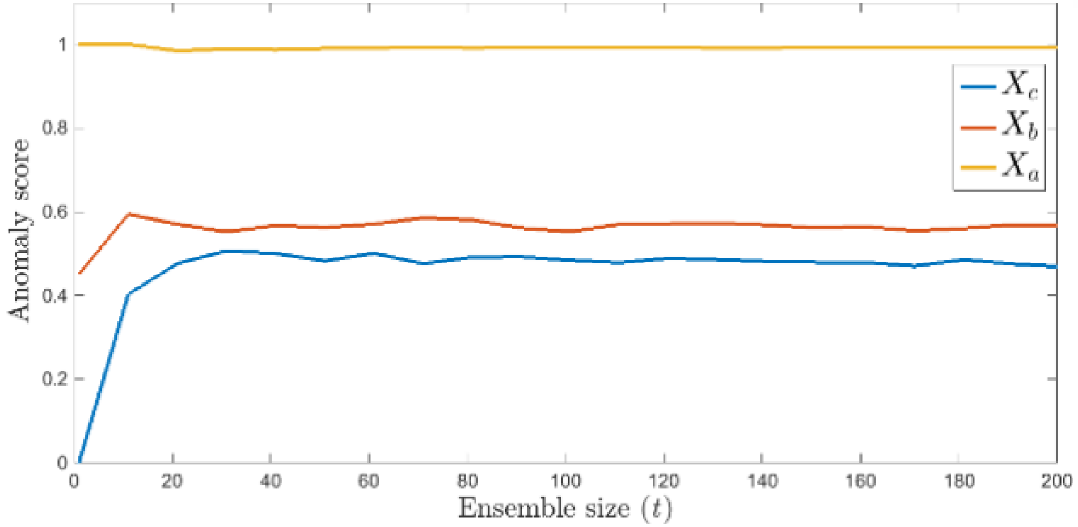
https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

[Bandaragoda et al, 2018]

Numerical example

- X_a has the maximum anomaly score
- X_b has a lower anomaly score
- X_c has the lowest anomaly score





https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

[Bandaragoda et al, 2018]

Structured data anomaly detection

Classic approach revisited

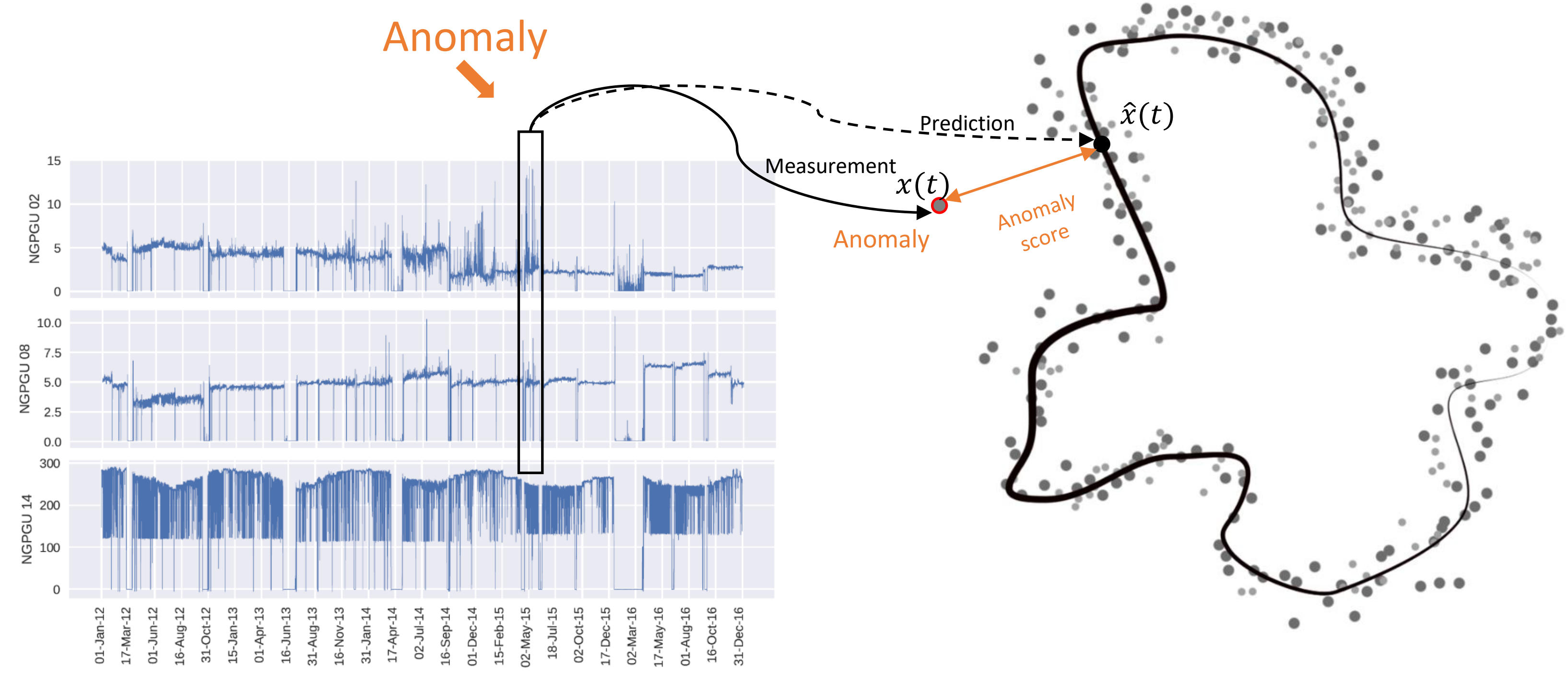
- A sample $D = \{\mathbf{x}_i\}_{i=1}^{n}$ is given, each $\mathbf{x} \in \mathbb{R}^d$.
- Construct models

$$\begin{aligned}\hat{x}_1 &= f_1(x_2, x_3, \dots, x_d),\\ \dots\\ \hat{x}_d &= f_d(x_1, x_2, \dots, x_{d-1}).\end{aligned}$$

- We have d anomaly scores for $\mathbf{x} = \{x_1, x_2, \dots, x_d\}$:

$$s_i(\mathbf{x}) = |\hat{x}_i - x_i|, i = \overline{1, d}.$$

Autoencoder anomaly detection. General approach



Local history

Embedding

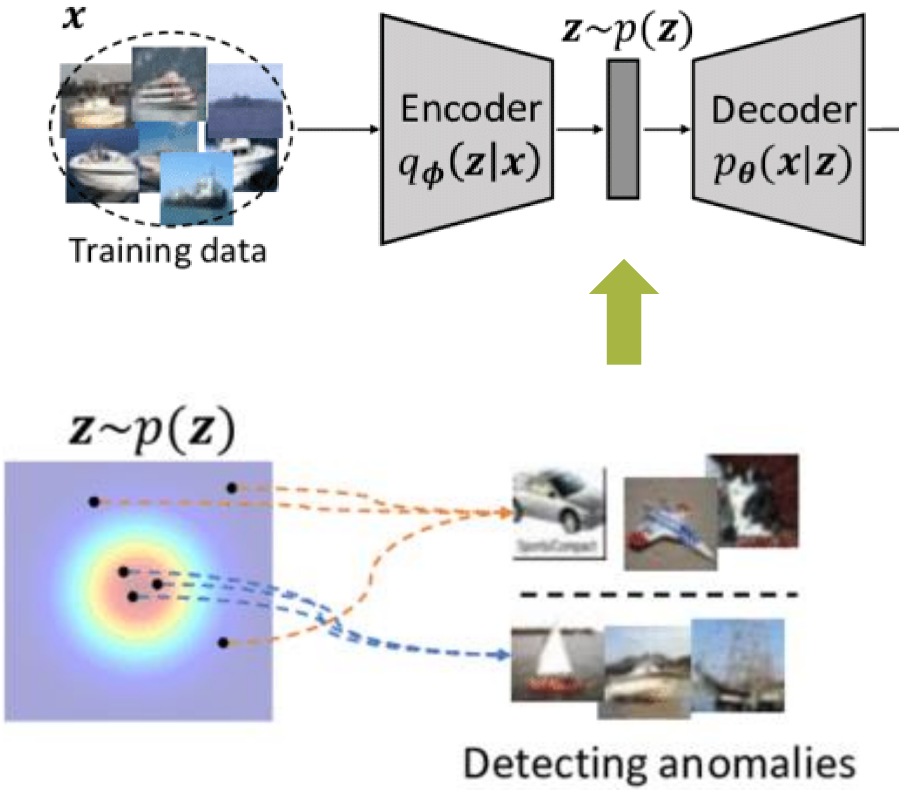
Unsupervised anomaly detection. General approach

- A sample $D = \{\mathbf{x}_i\}_{i=1}^{n}$ is given, each $\mathbf{x} \in \mathbb{R}^d$.
- Construct encoder and decoder model

$$
\begin{aligned}
\mathbf{z}_i &= e(\mathbf{x}_i),\\
\mathbf{x}_i \approx \hat{\mathbf{x}}_i &= d(\mathbf{z}_i) = d(e(\mathbf{x}_i)).
\end{aligned}
$$

- We have an anomaly score $s(\mathbf{x})$ for any $\mathbf{x}$:

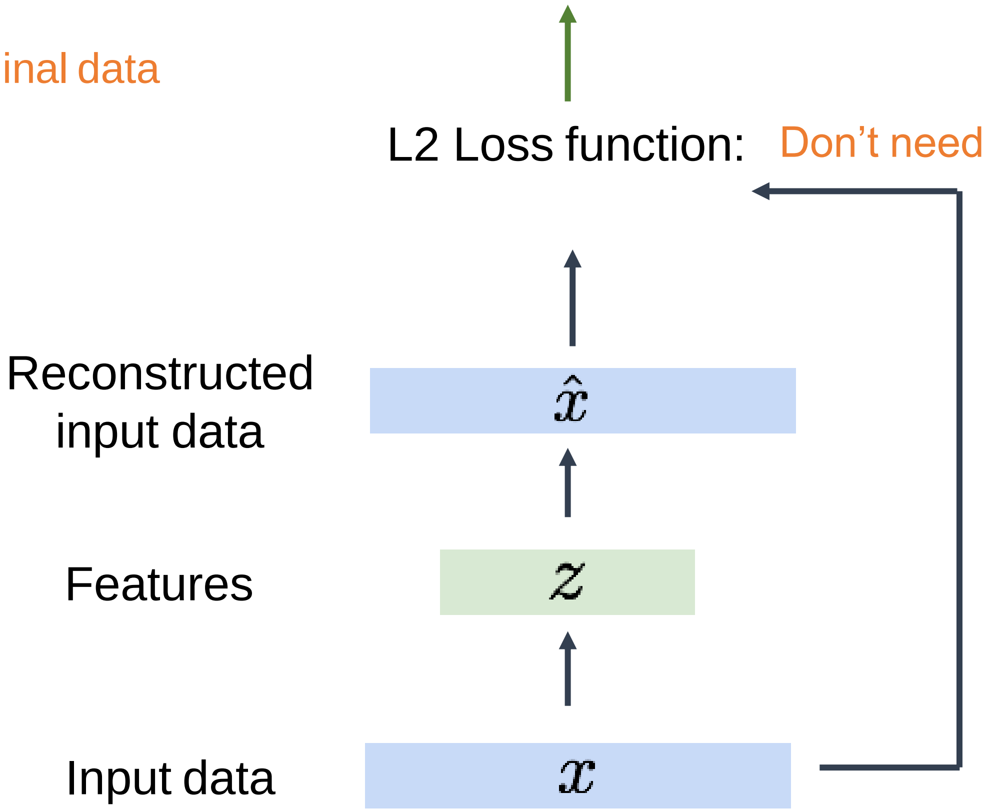
$$s(\mathbf{x}) = \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|.$$


$\boldsymbol{x}$
Training data
Encoder
$q_\phi(\mathbf{z}|\boldsymbol{x})$
$\mathbf{z}\sim p(\mathbf{z})$
Decoder
$p_\theta(\boldsymbol{x}|\mathbf{z})$
$\mathbf{z}\sim p(\mathbf{z})$
Detecting anomalies

Encoder, decoder examples: PCA, Autoencoder

Autoencoder. General idea

Train such that features can be used to reconstruct original data



If $\|x - \hat{x}\|^2 > \tau$
=> anomaly!

L2 Loss function: Don't need labels!

Reconstructed input data $\hat{x}$

Features z

Input data x

Reconstructed data

Encoder: 4-layer conv
Decoder: 4-layer upconv

Input data

Slides were adapted from lecture by Fei-Fei Li & Justin Johnson & Serena Yeung

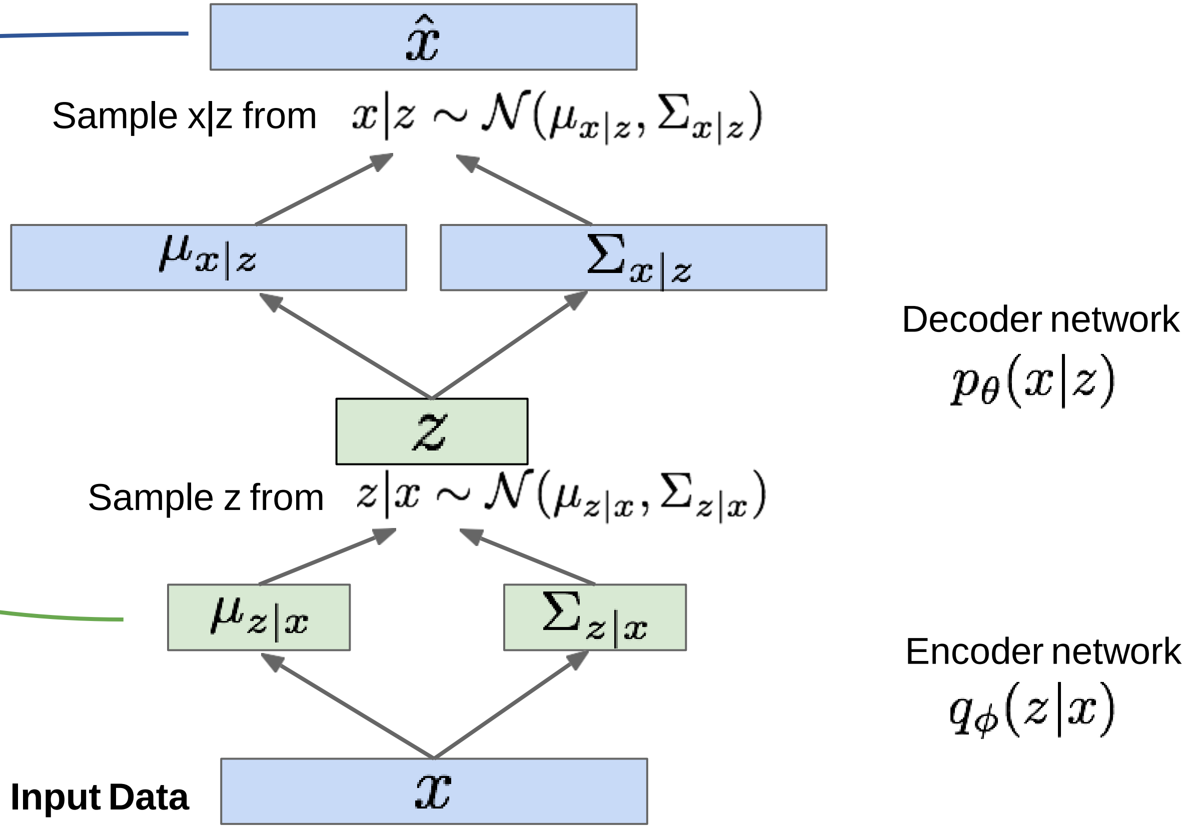
Variational Autoencoder

We maximize the likelihood lower bound

Maximize likelihood of original input being reconstructed

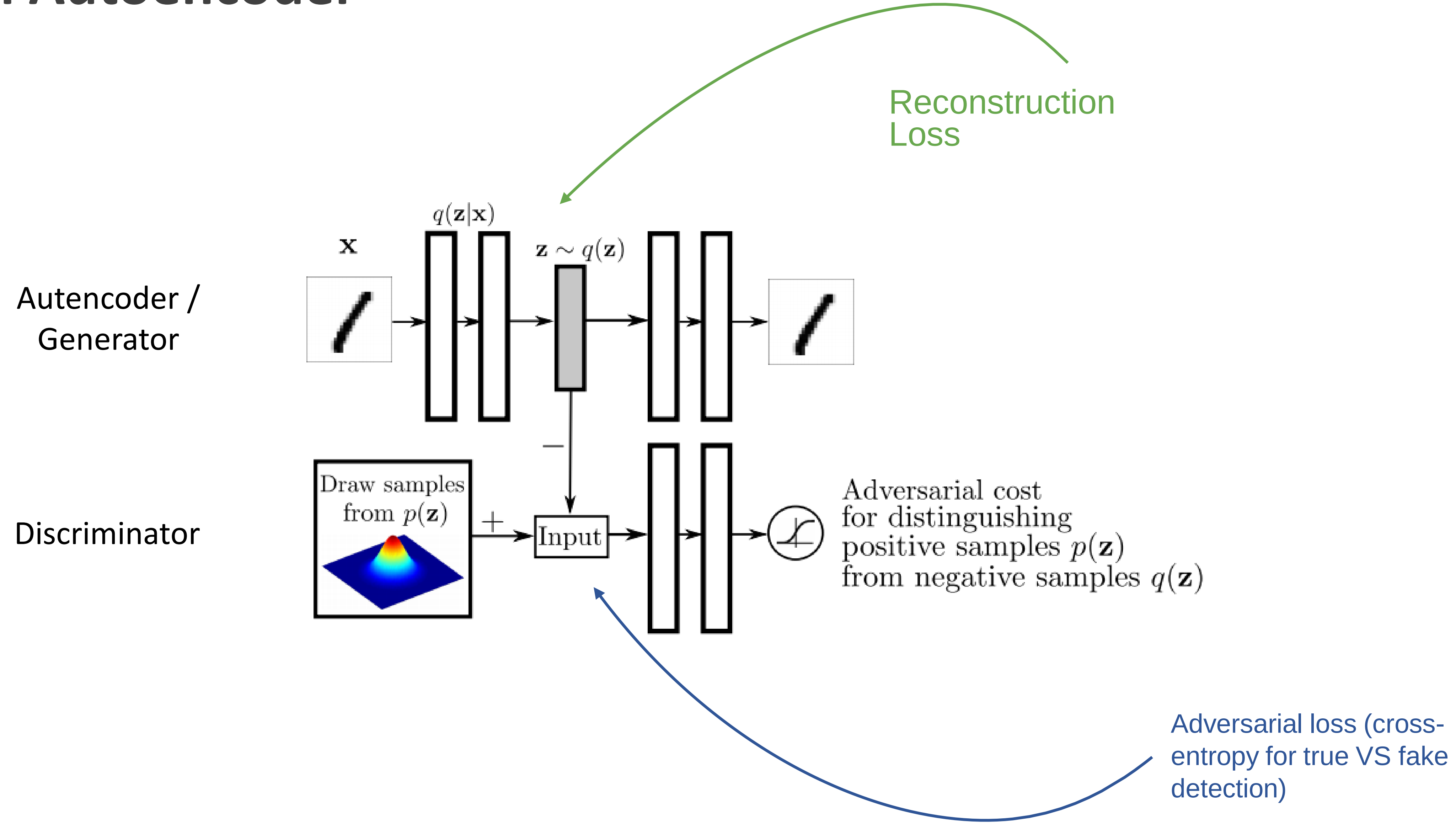
$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Make approximate posterior distribution close to prior



https://arxiv.org/pdf/1606.05908.pdf

Adversarial Autoencoder



Reconstruction Loss

Autencoder / Generator

Discriminator

$q(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$
$\mathbf{z} \sim q(\mathbf{z})$
$-$
Draw samples from $p(\mathbf{z})$
$+$
Input
Adversarial cost for distinguishing positive samples $p(\mathbf{z})$ from negative samples $q(\mathbf{z})$

Adversarial loss (cross-entropy for true VS fake detection)

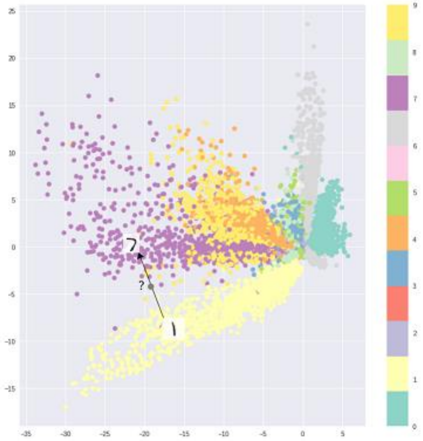
https://arxiv.org/pdf/1511.05644.pdf

Taxonomy of autoencoders

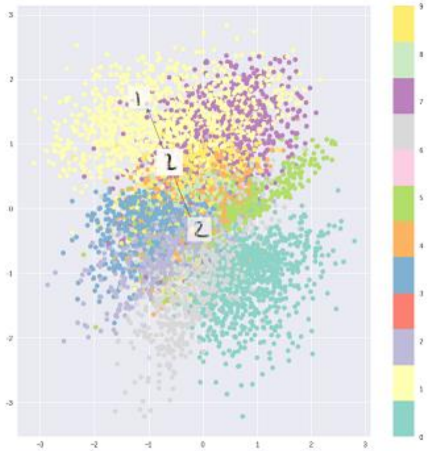
- **Just autoencoder**
  Sometimes we need just nonlinear PCA
  The latent space may not be continuous
  or allow easy interpolation.

- **Variational autoencoder**
  If you want precise control over your latent representations and
  what you would like them to represent, then choose VAE.
  Sometimes, precise modeling can capture better representations

- **Adversarial autoencoder**



AE



VAE

https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

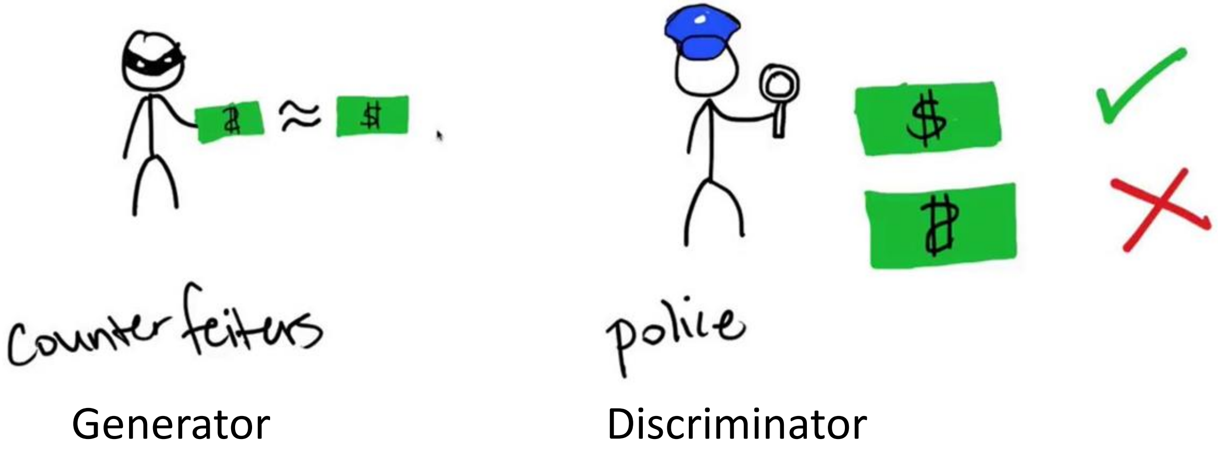
GANs for Anomaly Detection

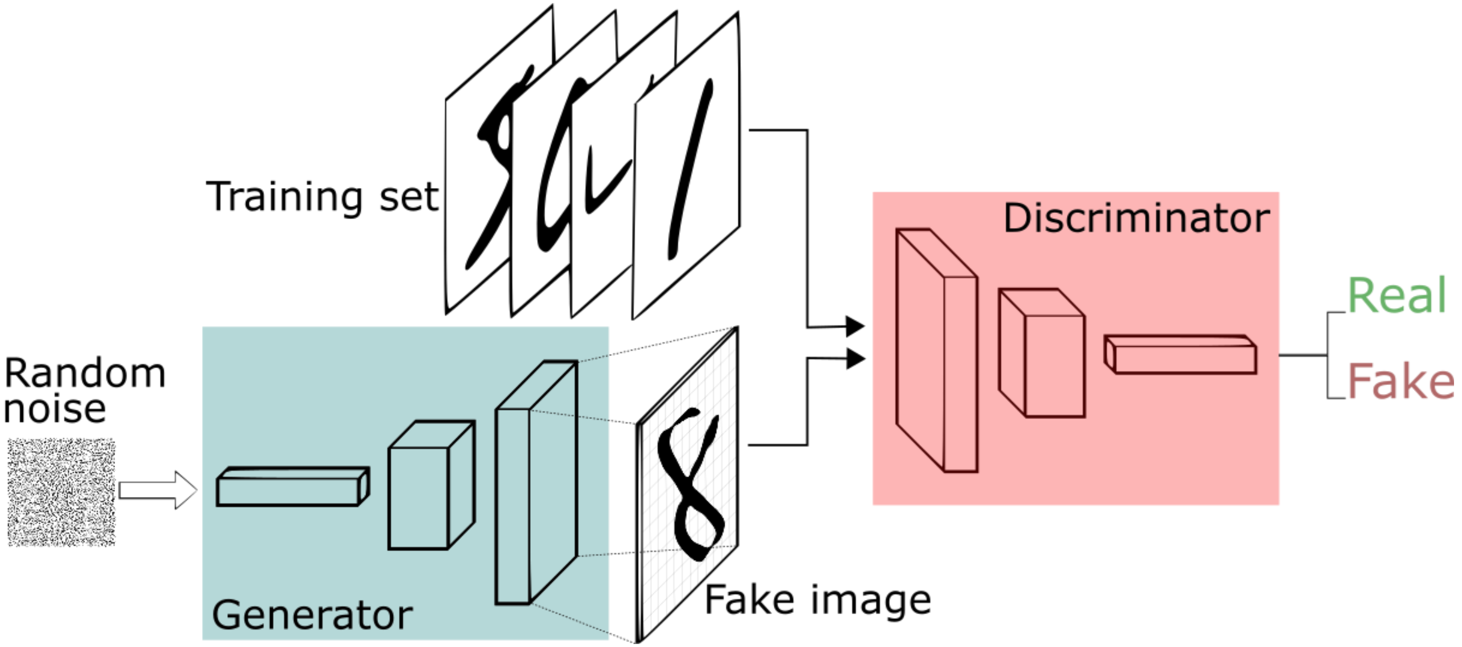
Generative Adversarial Network

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game!**





Generative Adversarial Network: formulas

Minimax objective function:

Discriminator outputs likelihood in (0,1) of real image

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

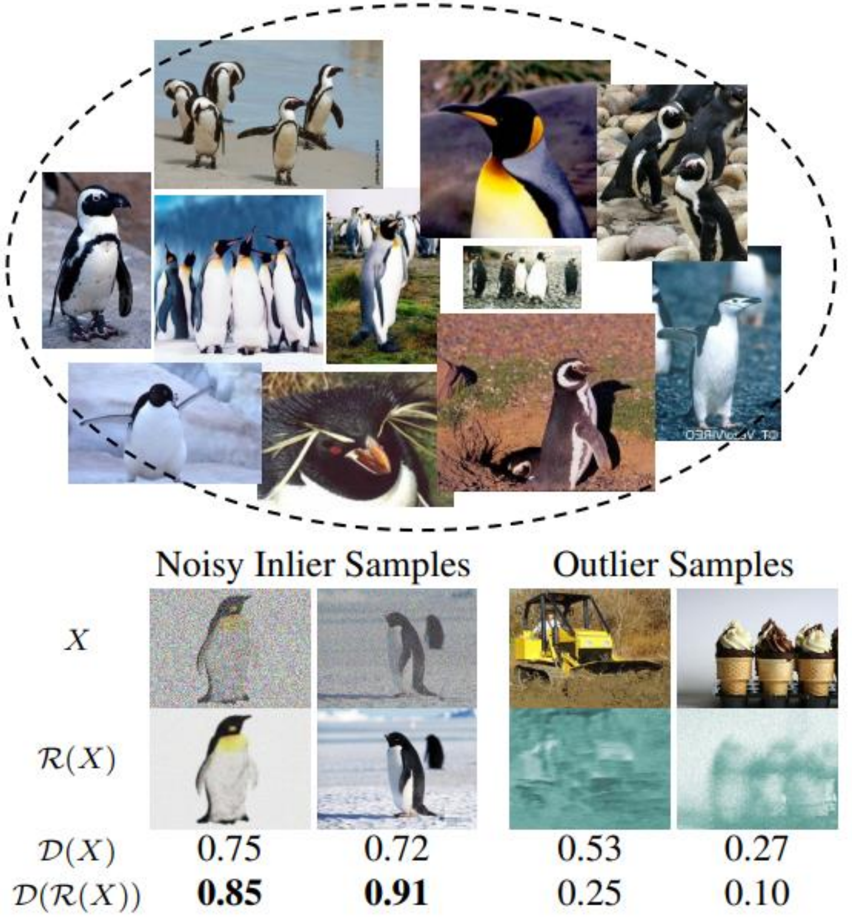
Discriminator output for real data x

Discriminator output for generated fake data G(z)

- *Discriminator* with parameters θ_d wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- *Generator* with parameters θ_g wants to **minimize objective** such that D(G(z)) is close to 1: the discriminator is fooled into thinking generated G(z) is real

GAN: example for novelty detection

The model is trained using images of penguins

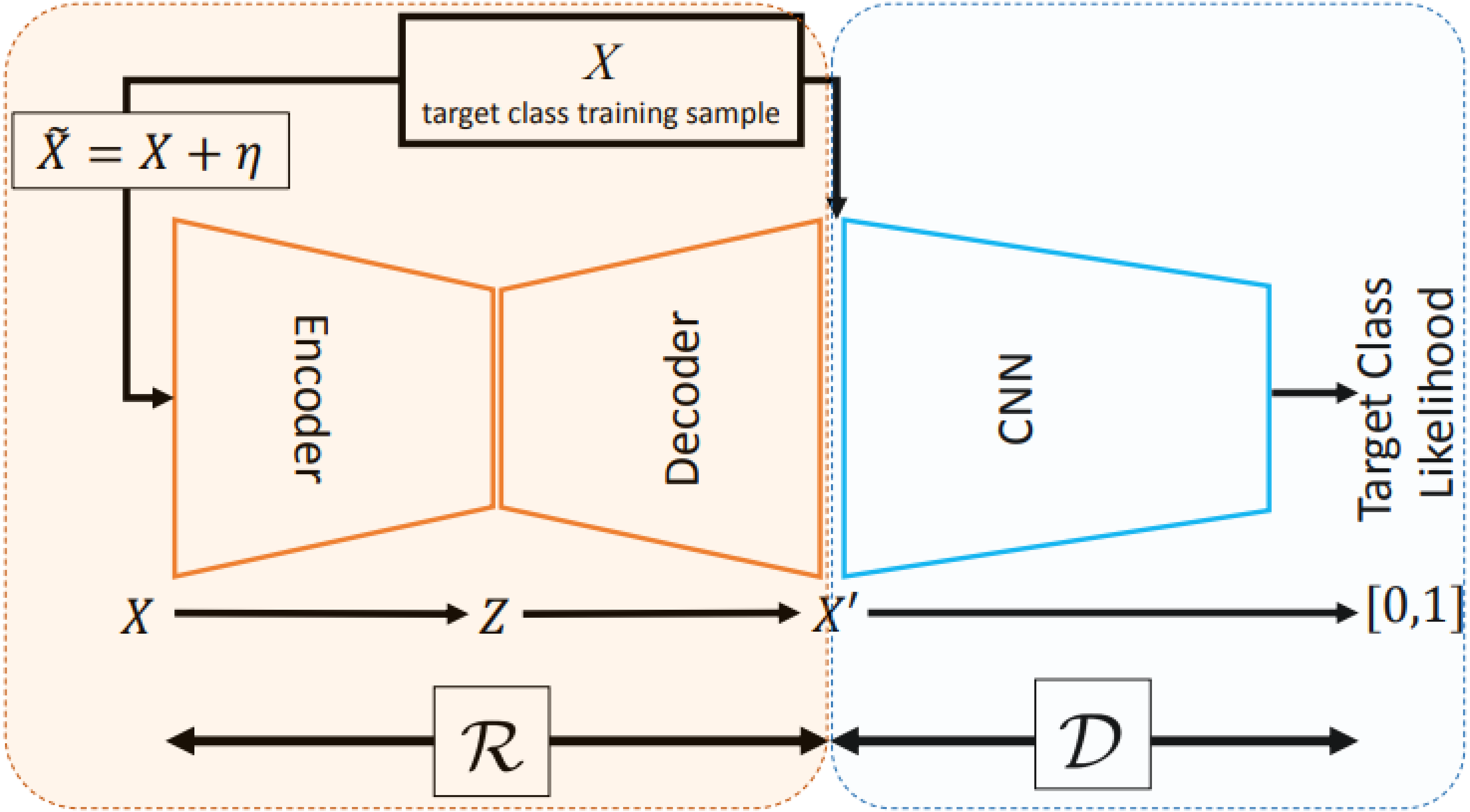

Noisy Inlier Samples
Outlier Samples
X
$\mathcal{R}(X)$
$\mathcal{D}(X)$ 0.75 0.72 0.53 0.27
$\mathcal{D}(\mathcal{R}(X))$ **0.85** **0.91** 0.25 0.10

- If we use noisy inliers and pass them to “autoencoder” ***R*** NN, we get enhanced images as the output
- If we use outlier sample instead, the output of ***R*** is distorted

M. Sabokrou et al. *Adversarially Learned One-Class Classifier for Novelty Detection*, CVPR, 2018

GAN: example for novelty detection

Architecture



Adversarial training

$$\mathcal{L} = \mathcal{L}_{\mathcal{R}+\mathcal{D}} + \lambda \mathcal{L}_{\mathcal{R}}$$

GAN loss

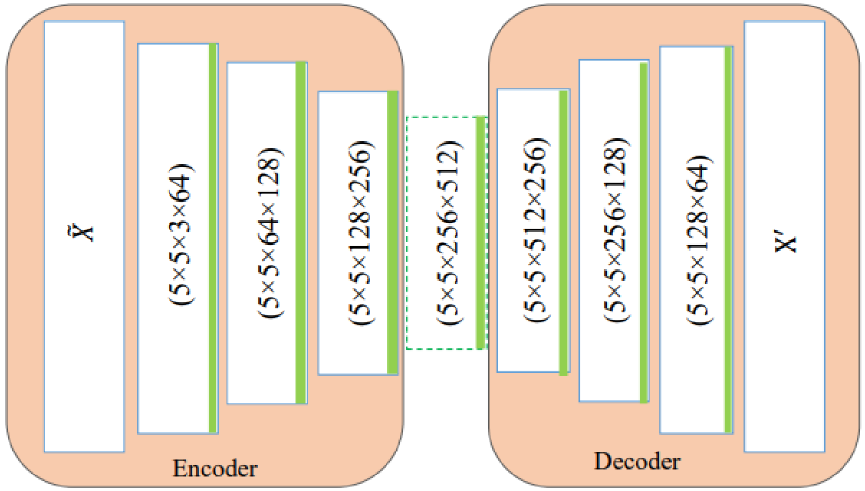
$$\mathcal{L}_{\mathcal{R}} = \|X - X'\|^2$$

Autoencoder generator

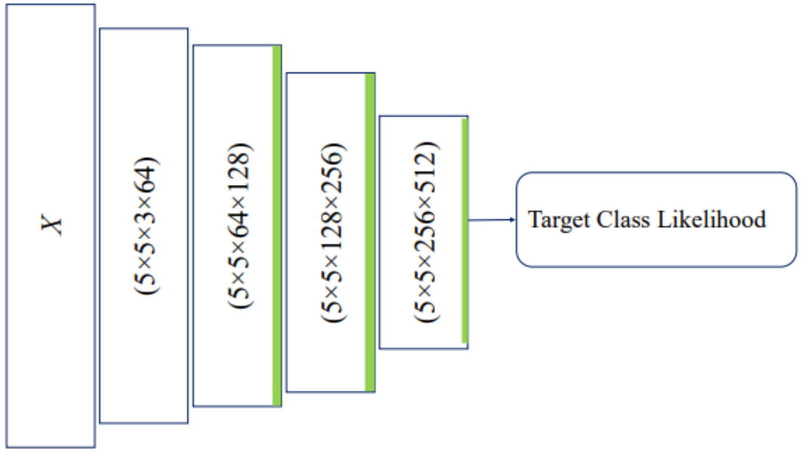
Discriminator gives anomaly score

https://arxiv.org/abs/1802.09088

Internal architectures



Autoencoder generator



Discriminator gives anomaly score

https://arxiv.org/abs/1802.09088

GAN: example for novelty detection

Anomaly score with state-of-the-art performance:

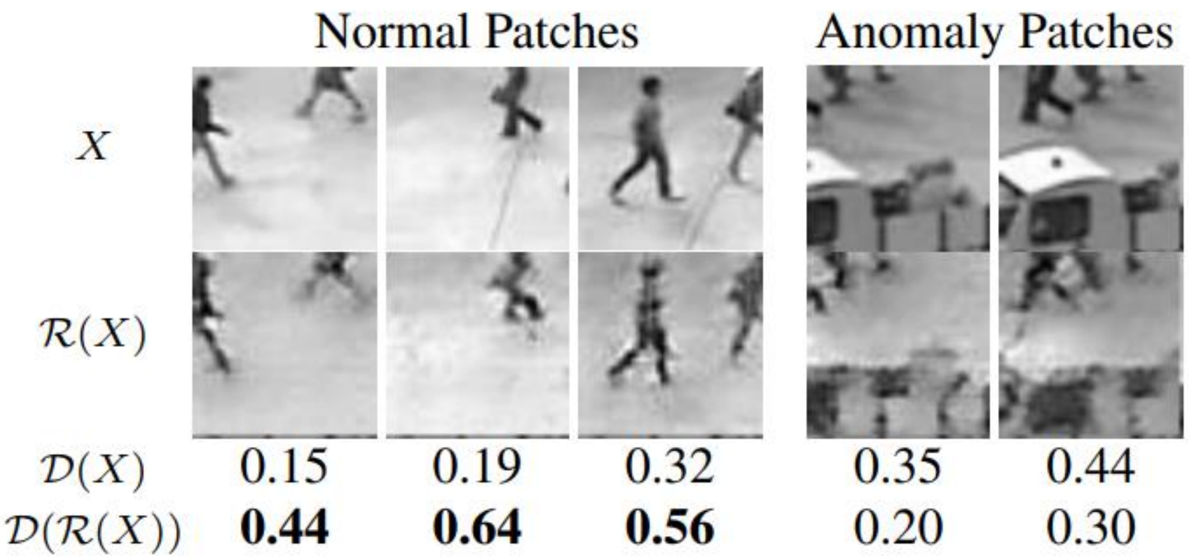
$$\mathrm{OCC}_1(X) = \begin{cases} \text{Target Class} & \text{if } \mathcal{D}(X) > \tau, \\ \text{Novelty (Outlier)} & \text{otherwise,} \end{cases}$$

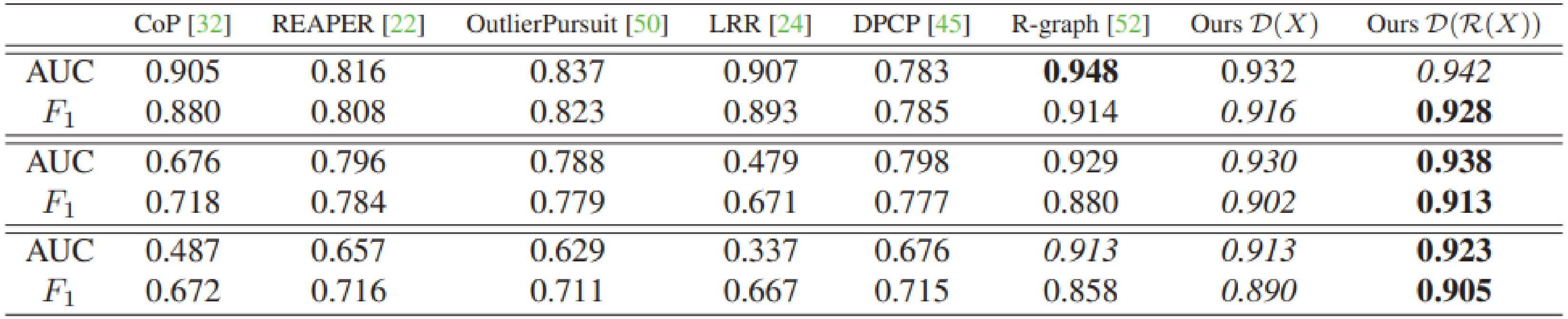
PCA, VAE, AAE

Anomaly score that utilizes encoder-decoder

$$\mathrm{OCC}_2(X) = \begin{cases} \text{Target Class} & \text{if } \mathcal{D}(\mathcal{R}(X)) > \tau, \\ \text{Novelty (Outlier)} & \text{otherwise.} \end{cases}$$

Model quality

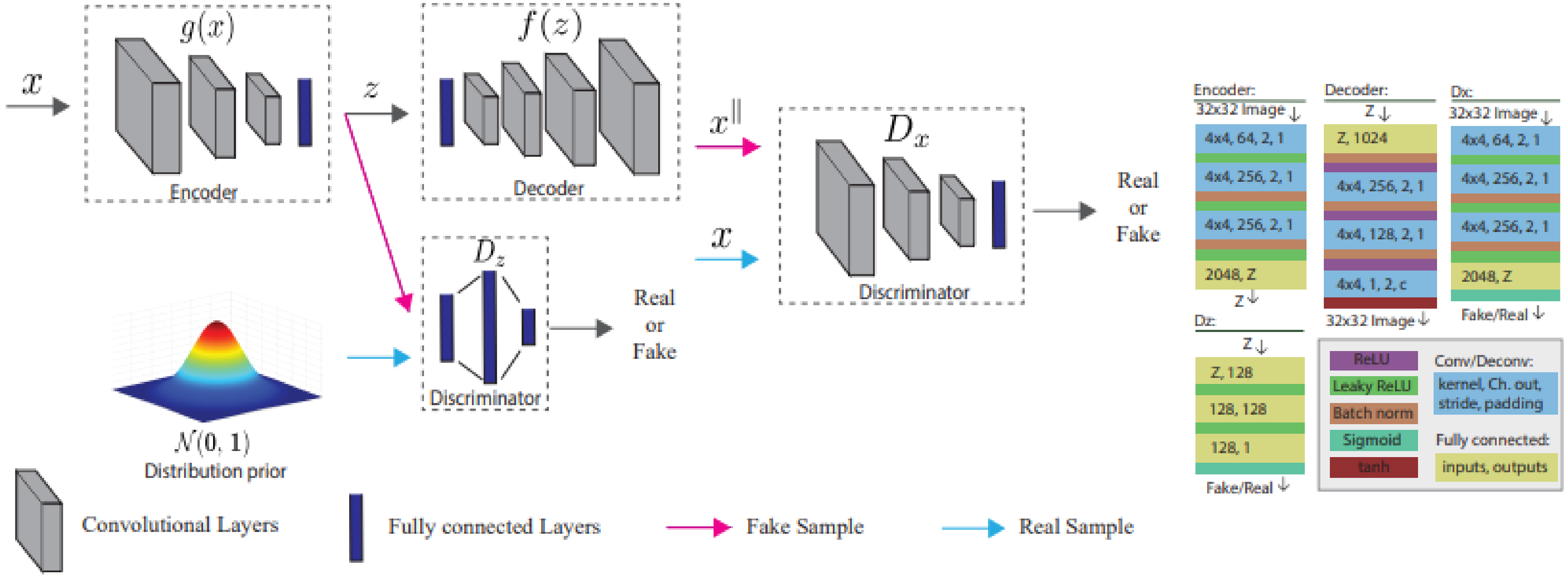




	CoP [32]	REAPER [22]	OutlierPursuit [50]	LRR [24]	DPCP [45]	R-graph [52]	Ours $\mathcal{D}(X)$	Ours $\mathcal{D}(\mathcal{R}(X))$
AUC	0.905	0.816	0.837	0.907	0.783	**0.948**	0.932	*0.942*
F_1	0.880	0.808	0.823	0.893	0.785	0.914	*0.916*	**0.928**
AUC	0.676	0.796	0.788	0.479	0.798	0.929	*0.930*	**0.938**
F_1	0.718	0.784	0.779	0.671	0.777	0.880	*0.902*	**0.913**
AUC	0.487	0.657	0.629	0.337	0.676	*0.913*	*0.913*	**0.923**
F_1	0.672	0.716	0.711	0.667	0.715	0.858	*0.890*	**0.905**

Adversarial autoencoders help

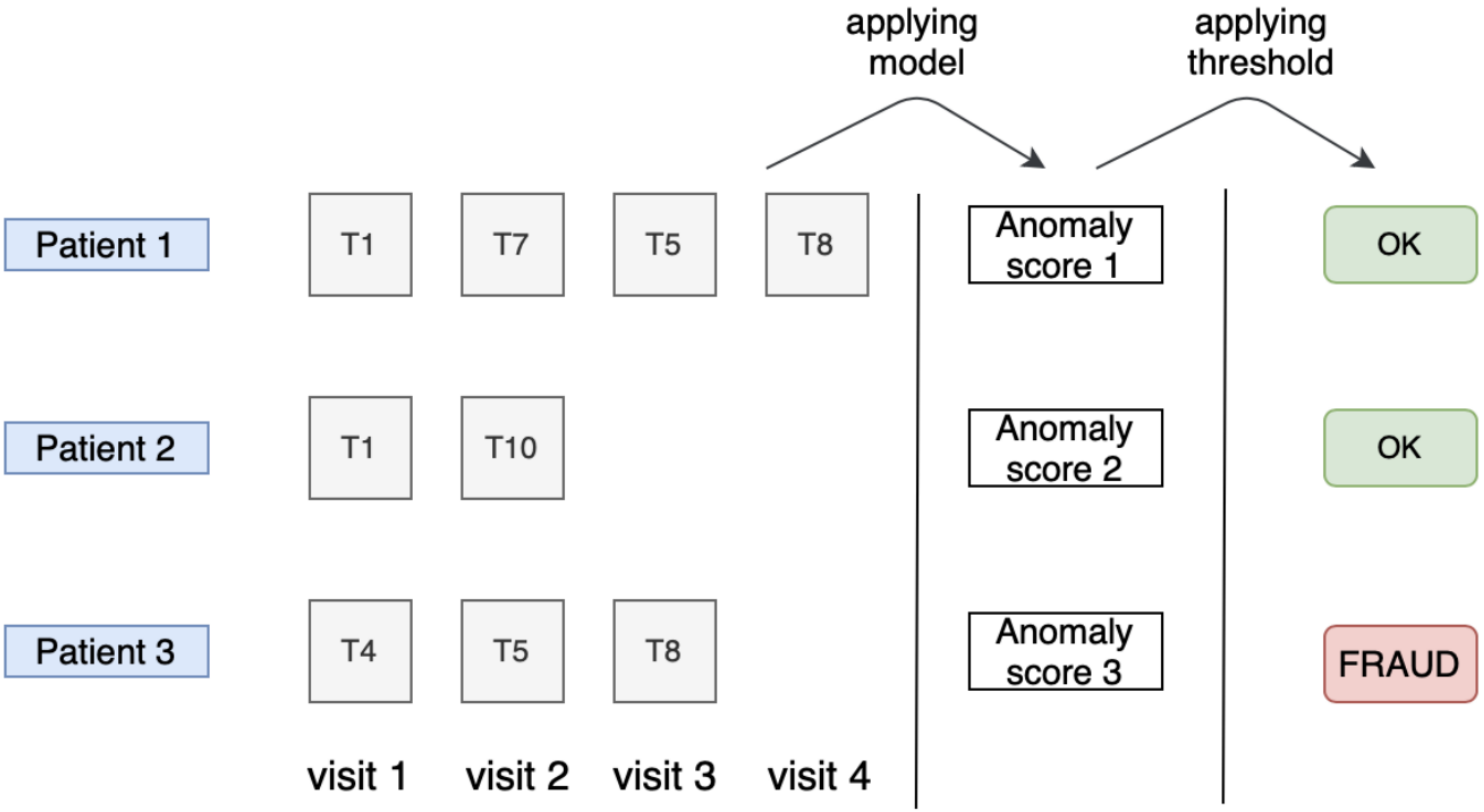
- Construct anomaly score s(x) using data
- Signal about anomaly if anomaly score is greater than some threshold t



https://papers.nips.cc/paper/7915-generative-probabilistic-novelty-detection-with-adversarial-autoencoders.pdf

Anomaly Detection for Time Series

Anomaly detection for sequential data: healthcare insurance



applying model

applying threshold

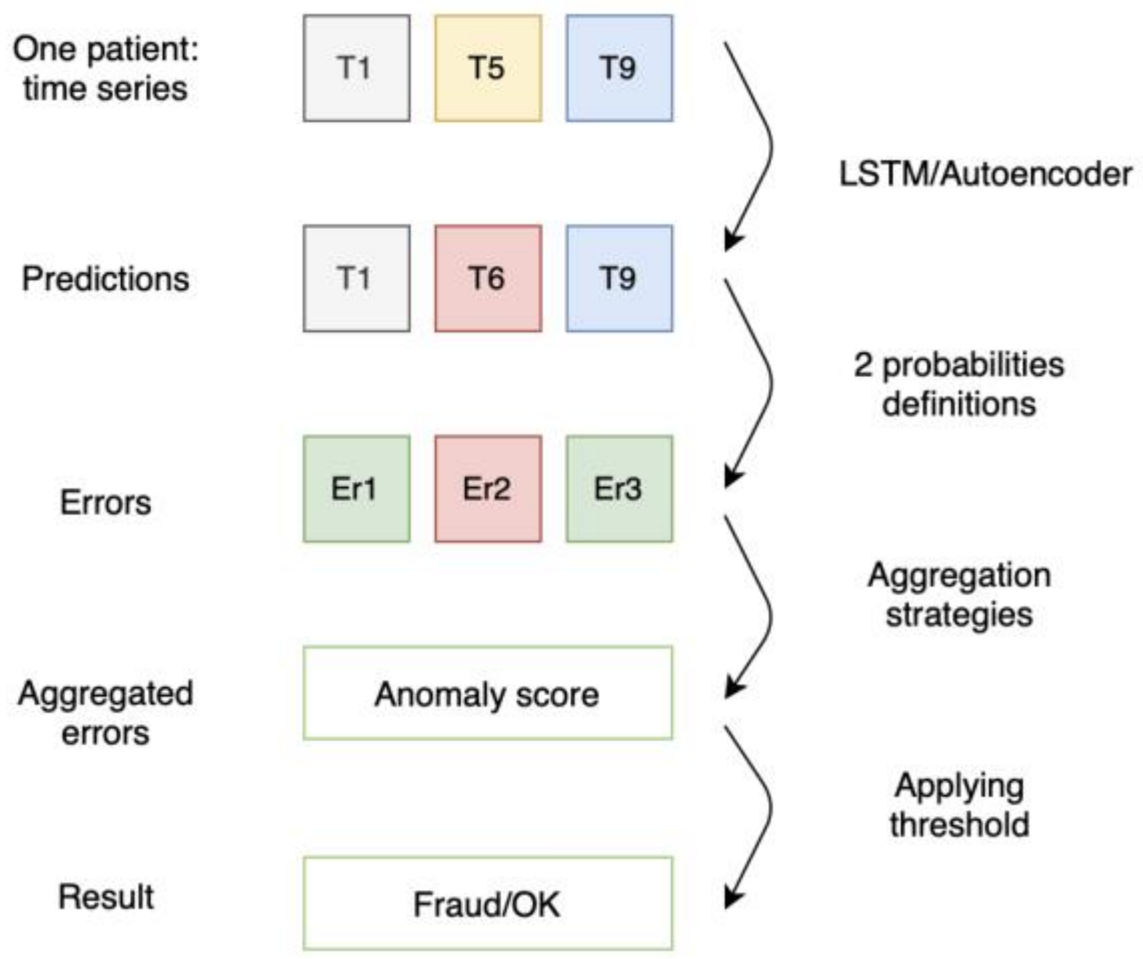
Patient 1 T1 T7 T5 T8 Anomaly score 1 OK

Patient 2 T1 T10 Anomaly score 2 OK

Patient 3 T4 T5 T8 Anomaly score 3 FRAUD

visit 1 visit 2 visit 3 visit 4

Anomaly detection for sequential data: healthcare insurance



Anomaly and novelty for Time Series

Problems:

1. SotA techniques of anomaly detection are rarely used for classic time series

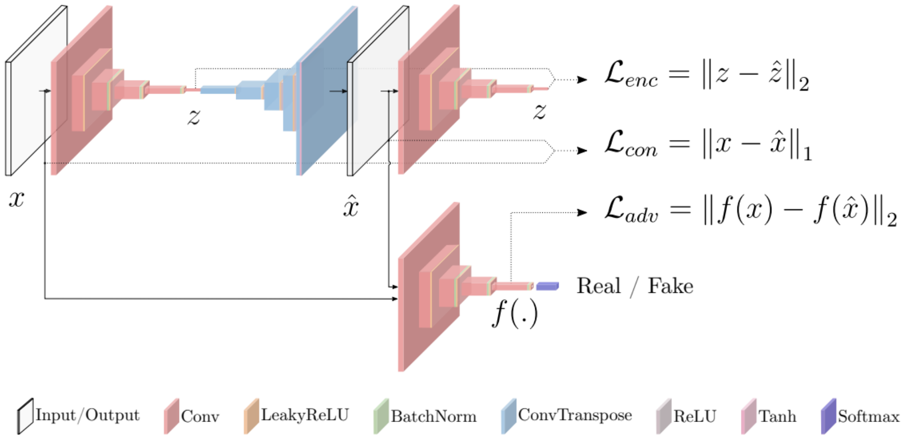
2. Available solutions don’t take into account the statistical nature of Time Series

Proposed solution:

1. To develop a loss function for GAN-based anomaly detection for time series

2. To take into account requirements of statistical change point detection models: low number of false alarms and small detection delay

3. To develop new resampling techniques by learning data distribution



x
z
$\hat{x}$
z
$f(.)$
$\mathcal{L}_{enc} = \|z - \hat{z}\|_2$
$\mathcal{L}_{con} = \|x - \hat{x}\|_1$
$\mathcal{L}_{adv} = \|f(x) - f(\hat{x})\|_2$
Real / Fake
Input/Output
Conv
LeakyReLU
BatchNorm
ConvTranspose
ReLU
Tanh
Softmax

Li D. et al. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks // arXiv:1901.04997. – 2019.

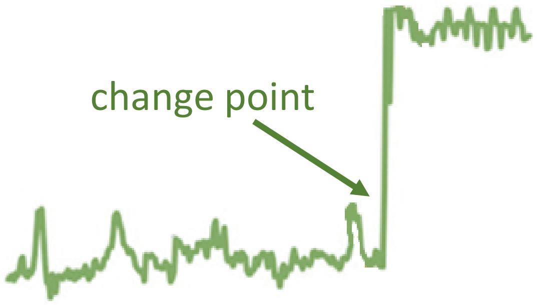
Change detection in semi-structured data

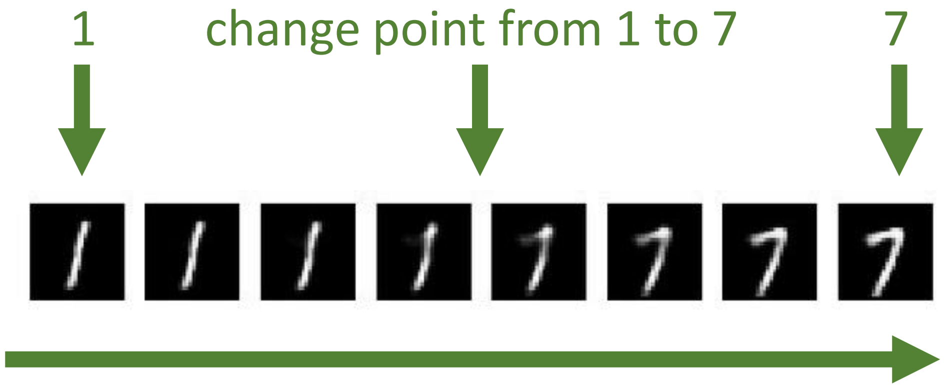
The change-point detection (CPD) model signals about time of change in the data distribution

Semi-structured data – sequences of semi-structured data (images, texts)

Goal: minimize Detection Delay & minimize number of False Alarms

Problems: Can’t apply classic method for semi-structured data



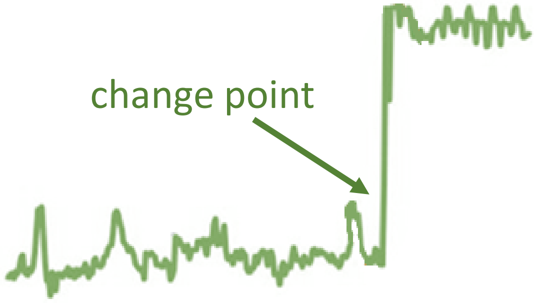


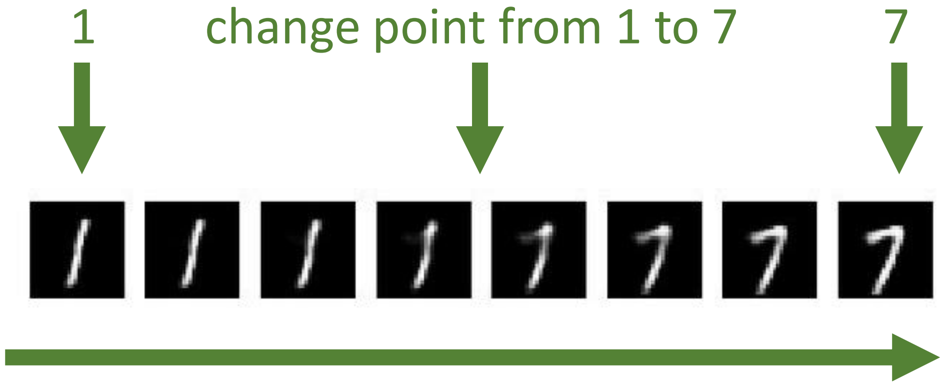
A sequence of MNIST images is an example of semi-structured data

Change detection in semi-structured data

Proposed solution:

1. Develop a data embedding procedure
2. End2end methods based on statistical tests or detection outliers in embedded space – unsupervised anomaly detection
3. Develop new loss function for direct minimization of the problem specific metrics

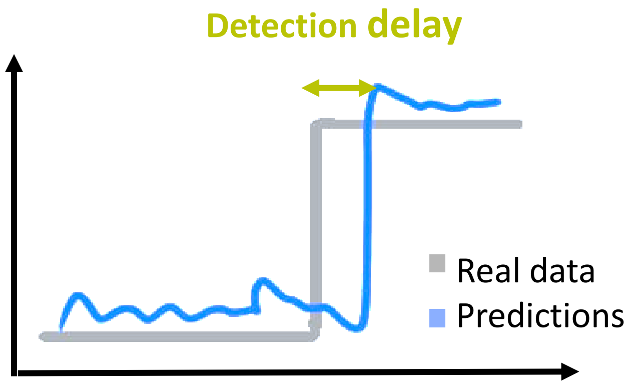


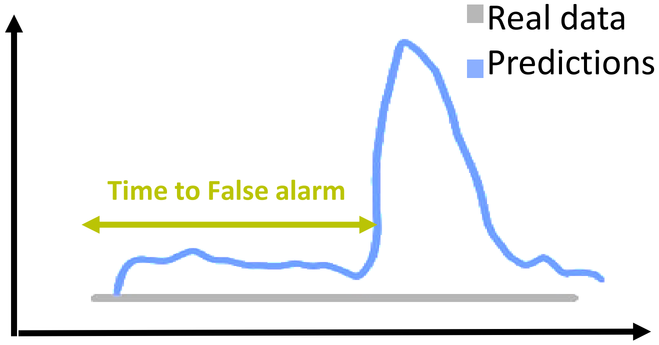


A sequence of MNIST images is an example of semi-structured data

Our end2end approach

We concentrate on typical quality metrics for change-point detection: delay detection and mean time to False alarm.





We optimize lower bounds for these metrics:

- p_k is the model’s change point probability at moment k,
- T – hyperparameter that restricts the length of the considered sequence.

$$Loss_{detection\_delay} = \sum_{t=\theta}^{T}(t - \theta)\, p_t \prod_{k=\theta}^{t-1}(1 - p_k) + (T + 1)\prod_{k=\theta}^{T}(1 - p_k),$$

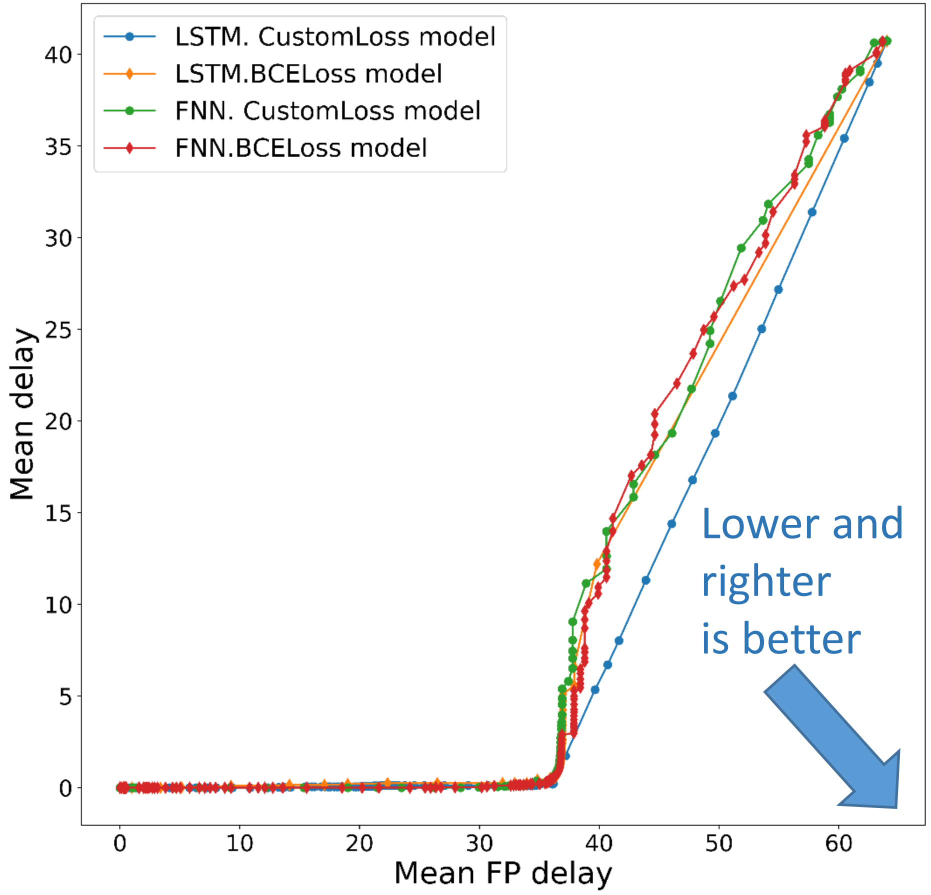
$$Loss_{FP\_delay} = 1 - \sum_{t=0}^{\theta}(t - \theta)p_t \prod_{k=0}^{\theta}(1 - p_k)$$

Results

Dataset: sequences with images from MNIST. There are sequences with (e.g. from 1 to 7) and without (e.g. from 1 to 1) change point.

We compare LSTM and fully connected neural network (FNN) architectures, as well as binary cross entropy loss (BCELoss) and our proposed loss.

LSTM with proposed loss function has a better Pareto frontier with respect to the mean detection delay and the mean false positive delay.



Lower and righter is better

References

Change point detection (CPD). Basic knowledge and main statistical approaches:

- Shiryaev A. N. Stochastic disorder problems. – Springer International Publishing, 2019.
- Romanenkova E. et al. Real-Time Data-Driven Detection of the Rock-Type Alteration During a Directional Drilling //IEEE Geoscience and Remote Sensing Letters. – 2019.

Supervised CPD:

- Malhotra P. et al. Long short term memory networks for anomaly detection in time series //ESANN proceedings, 2015.
- Hundman K. et al. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding //Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – 2018

Conclusions

Factors to consider when choosing an Anomaly Detector

- Few parameters
  - parameter-free the best
  - easy to tune; not too sensitive to parameter setting

- Fast runtime: can scale up to large datasets and high dimensional datasets

- Low space complexity

- Known behaviours under different data properties

- Can deal with different types of anomalies

- Its ability to deal with high dimensional problems

- Understand the nature of anomalies and the best match algorithm

https://federation.edu.au/__data/assets/pdf_file/0011/443666/ICDM2018-Tutorial-Final.pdf

Take-home messages

- Anomaly detection is a challenging problem
- Often problem-specific knowledge helps

- Common approaches are Autoencoder-based and Isolation forest
- There are some time-series specific approaches: the problem is similar to the change point detection problem

More references?

- Overview of anomaly detection for tabular data
  https://www.youtube.com/watch?v=12Xq9OLdQwQ

- A collection of *awesome* anomaly detection papers
  https://awesomeopensource.com/project/hoya012/awesome-anomaly-detection

- A collection of *awesome* anomaly detection resources
  https://github.com/yzhao062/anomaly-detection-resources

- Link-based list of anomaly detection methods
  https://github.com/zhuyiche/awesome-anomaly-detection