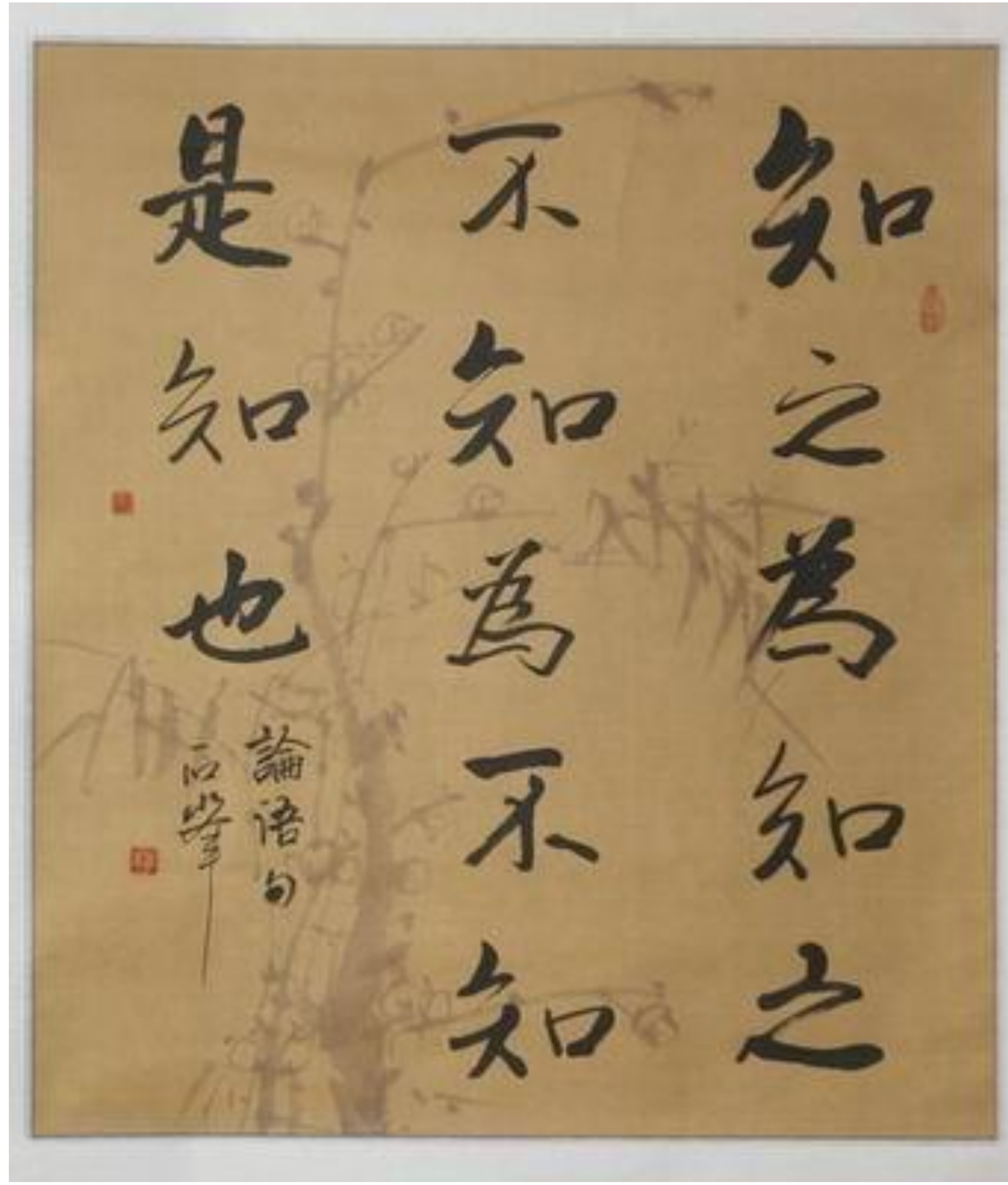
Anomaly Detection

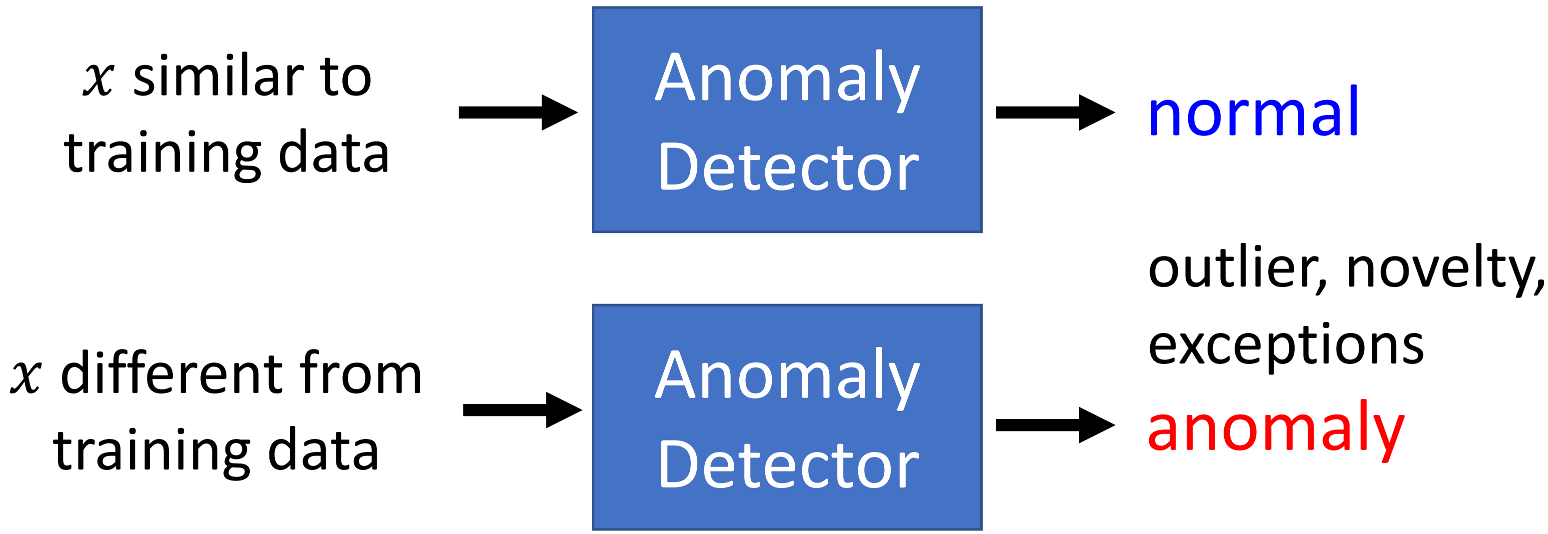
Hung-yi Lee
李宏毅



Source of image: https://zhidao.baidu.com/question/757382451569322644.html

Problem Formulation

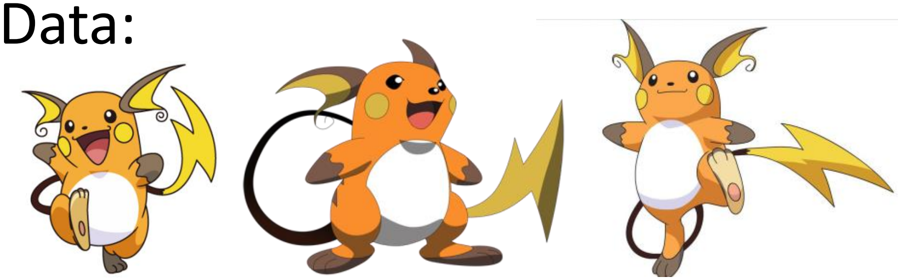
- Given a set of training data $\{x^1, x^2, \cdots, x^N\}$
- We want to find a function detecting input x is *similar* to training data or not.



Different approaches use different ways to determine the similarity.

What is Anomaly?

Training Data:



anomaly

Training Data:

anomaly

Training Data:

寶可夢
(神奇寶貝)

數碼寶貝

anomaly

Applications

- Fraud Detection
  - Training data: 正常刷卡行為, x: 盜刷？
    - Ref: https://www.kaggle.com/ntnu-testimon/paysim1/home
    - Ref: https://www.kaggle.com/mlg-ulb/creditcardfraud/home
- Network Intrusion Detection
  - Training data: 正常連線, x: 攻擊行為？
    - Ref: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
- Cancer Detection
  - Training data: 正常細胞, x: 癌細胞
    - Ref: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/home

Binary Classification?

- Given normal data $\{x^1, x^2, \cdots, x^N\}$ $\Rightarrow$ Class 1
- Given anomaly $\{\tilde{x}^1, \tilde{x}^2, \cdots, \tilde{x}^N\}$ $\Rightarrow$ Class 2
- Then training a binary classifier ……

Binary Classification?

Binary Classification?

- Given normal data $\{x^1, x^2, \cdots, x^N\}$ ➡ Class 1
- Given anomaly $\{\tilde{x}^1, \tilde{x}^2, \cdots, \tilde{x}^N\}$ ➡ Class 2
- Then training a binary classifier ……

x (Pokémon)

$\tilde{x}$ (NOT Pokémon)

cannot be considered as a class

Even worse, in some cases, it is difficult to find anomaly example ……

Categories

Training data: $\{x^1, x^2, \cdots, x^N\}$

With labels: $\{\hat{y}^1, \hat{y}^2, \cdots, \hat{y}^N\}$ → Classifier

The classifier can output "*unknown*" (none of the training data is labelled "unknown")

Open-set Recognition

Clean: All the training data is normal.

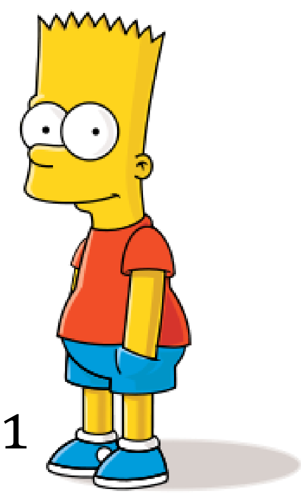
Polluted: A little bit of training data is anomaly.

Case 1: With Classifier

Example Application

- From The Simpsons or not

x



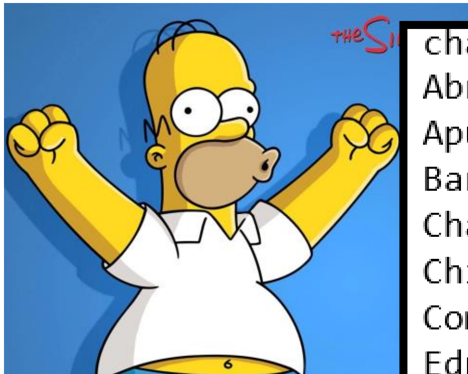
x^1

x^2

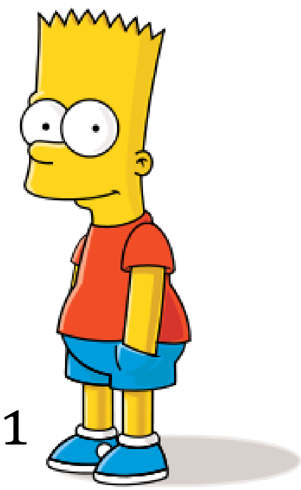
x^3

x^4

Source of model: https://www.kaggle.com/alexattia/the-simpsons-characters-dataset/

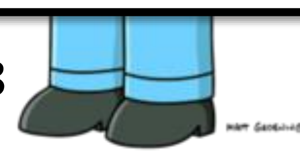


character	Precision	Support
Abraham Grampa Simpson	0.96	47
Apu Nahasapeemapetilon	0.98	49
Bart Simpson	0.94	51
Charles Montgomery Burns	0.92	48
Chief Wiggum	1.00	50
Comic Book Guy	0.98	48
Edna Krabappel	0.98	47
Homer Simpson	0.94	49
Kent Brockman	1.00	48
Krusty The Clown	0.98	51
Lisa Simpson	0.92	51
Marge Simpson	0.98	51
Milhouse Van Houten	0.94	51
Moe Szyslak	0.98	49
Ned Flanders	0.89	53
Nelson Muntz	0.98	45
Principal Skinner	0.91	55
Sideshow Bob	0.98	47
Total	0.96	890



x^1 $\hat{y}^1$ =霸子

x^2 $\hat{y}^2$ =麗莎



x^3 $\hat{y}^3$ = 荷馬

x^4 $\hat{y}^4$ = 美枝

How to use the Classifier

Anomaly Detection:

$$f(x) = \begin{cases} normal, & c(x) > \lambda \\ anomaly, & c(x) \le \lambda \end{cases}$$

How to estimate Confidence

→ Simpsons Character Classifier →

霸子	$\boxed{0.97}$
麗莎	0.01
荷馬	0.01
美枝	0.01

Very Confident

Normal

→ Simpsons Character Classifier →

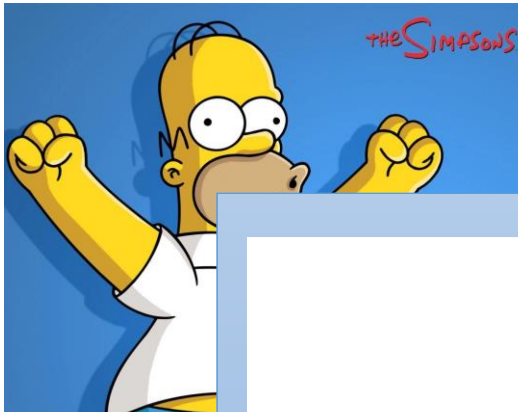
霸子	0.24
麗莎	$\boxed{0.26}$
荷馬	0.25
美枝	0.25

Not Confident

Anomaly

Confidence: the maximum scores

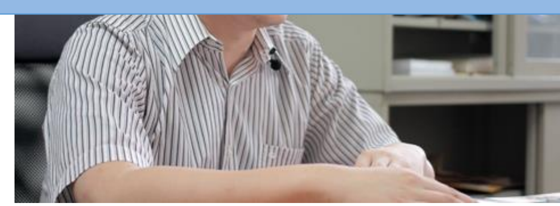
or negative Entropy



柯阿三 0.34

鹹 0.31

王 0.10



郭董 0.12

(辛普森家庭腳色)

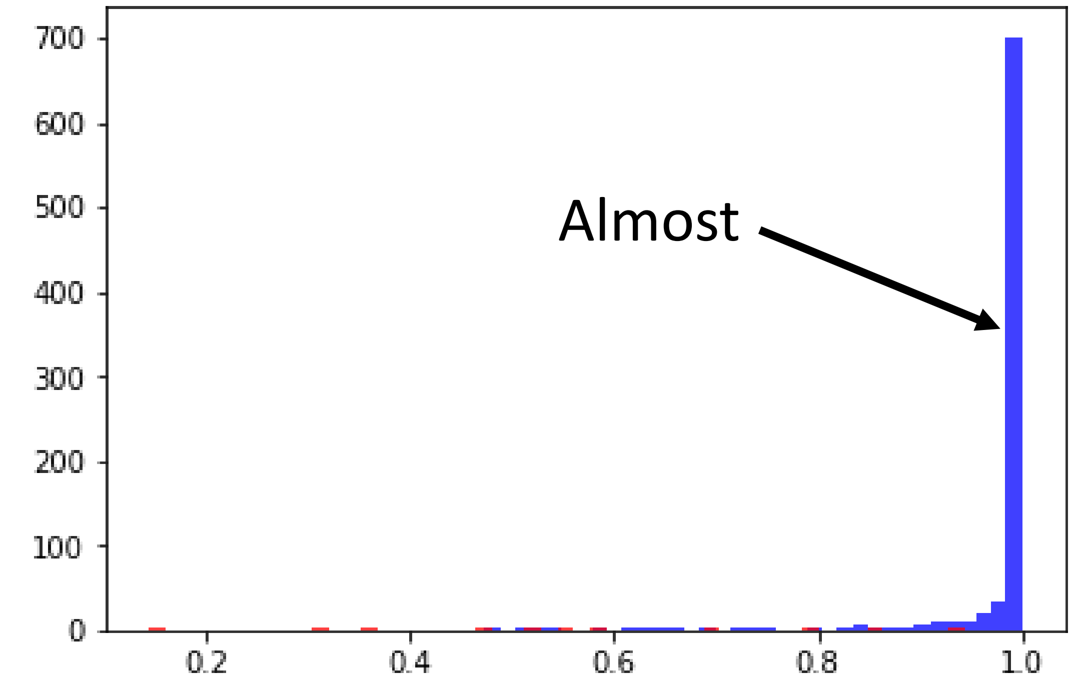
柯阿三 0.99

三 0.63

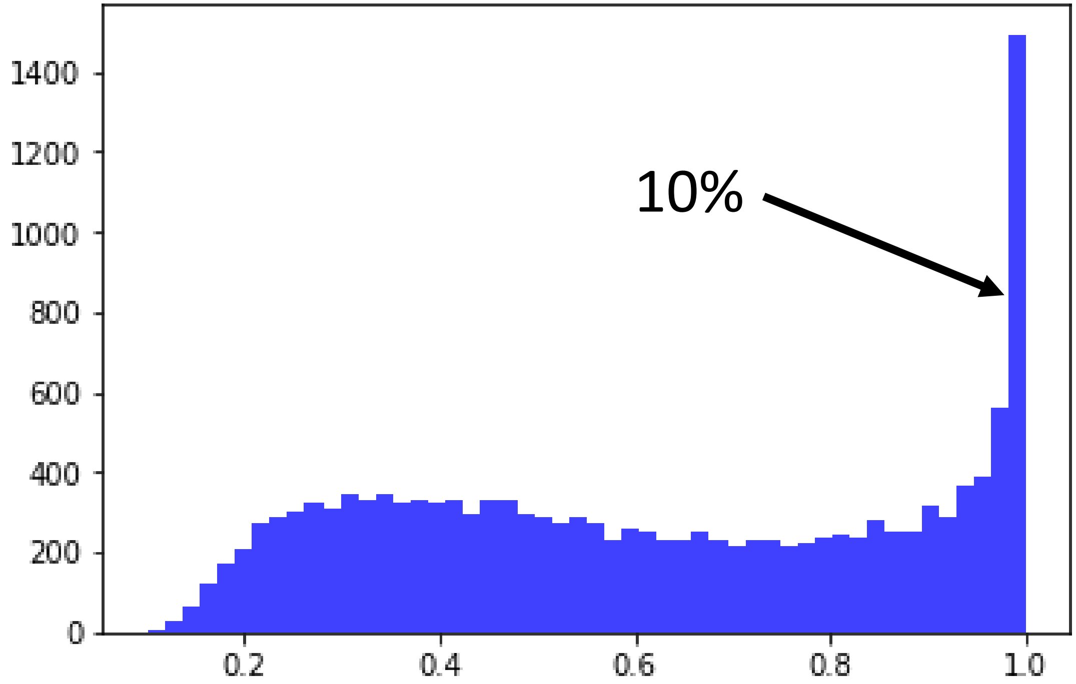
0.08

小丑阿基 0.04

孔龍金 0.03



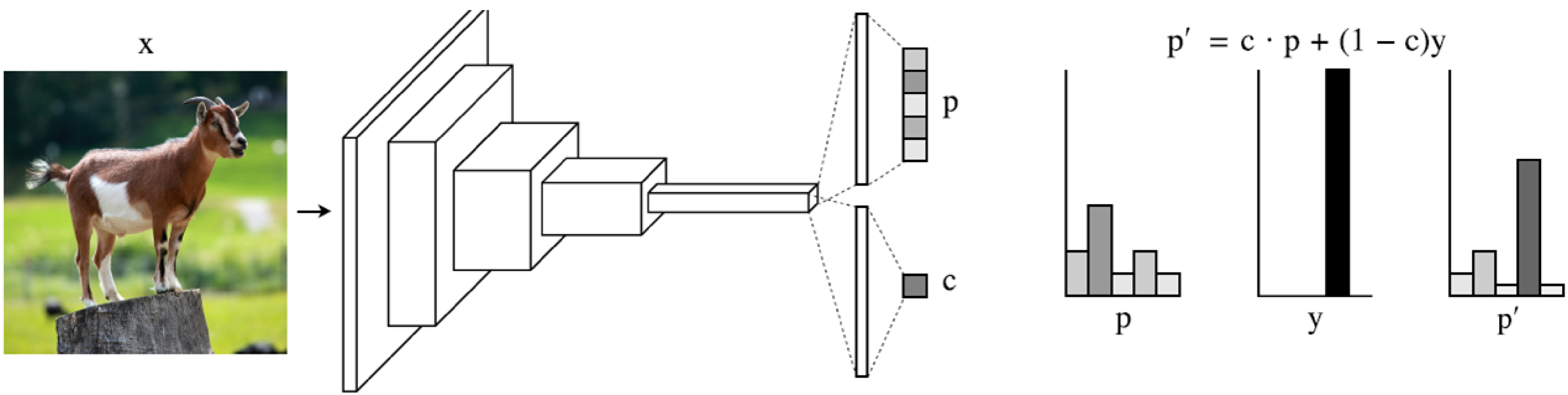
Confidence score distribution for ***characters from Simpsons***



Confidence score distribution for ***anime characters***

Outlook: Network for Confidence Estimation

- Learning a network that can directly output confidence



Terrance DeVries, Graham W. Taylor, Learning Confidence for Out-of-Distribution Detection in Neural Networks, arXiv, 2018

(not today)

Example Framework

Training Set: Images x of characters from Simpsons.

Each image x is labelled by its characters $\hat{y}$.

Train a classifier, and we can obtain confidence score $c(x)$ from the classifier.

$$f(x) = \begin{cases} normal, & c(x) > \lambda \\ anomaly, & c(x) \leq \lambda \end{cases}$$

Dev Set: Images x

Label each image x is from Simpsons or not.

We can compute the ***performance*** of $f(x)$

Using dev set to determine λ and other hyperparameters.

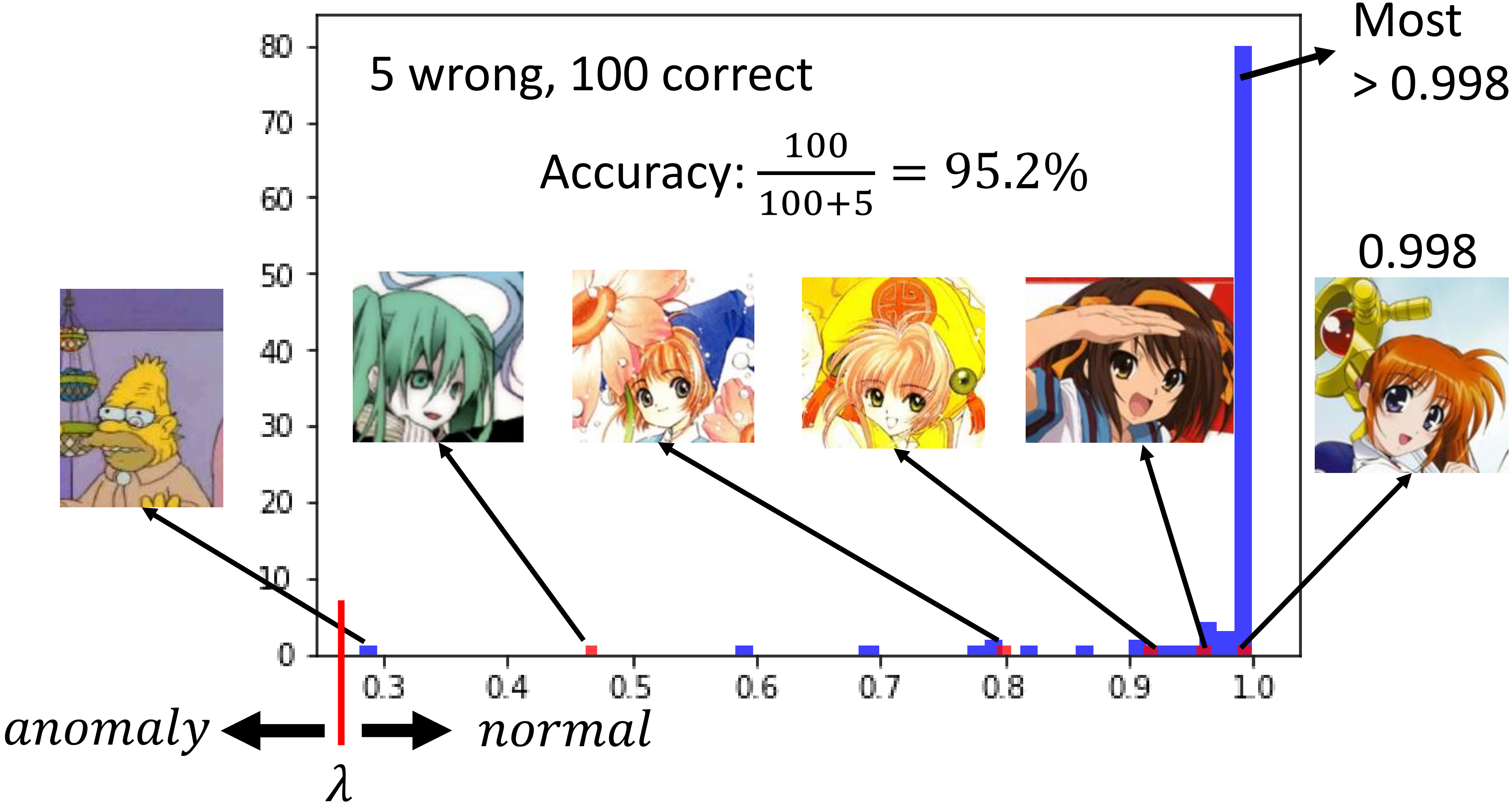
Testing Set: Images x ➡ from Simpsons or not

Evaluation

Accuracy is not a good measurement!

A system can have high accuracy, but do nothing.

100 Simpsons, 5 anomalies

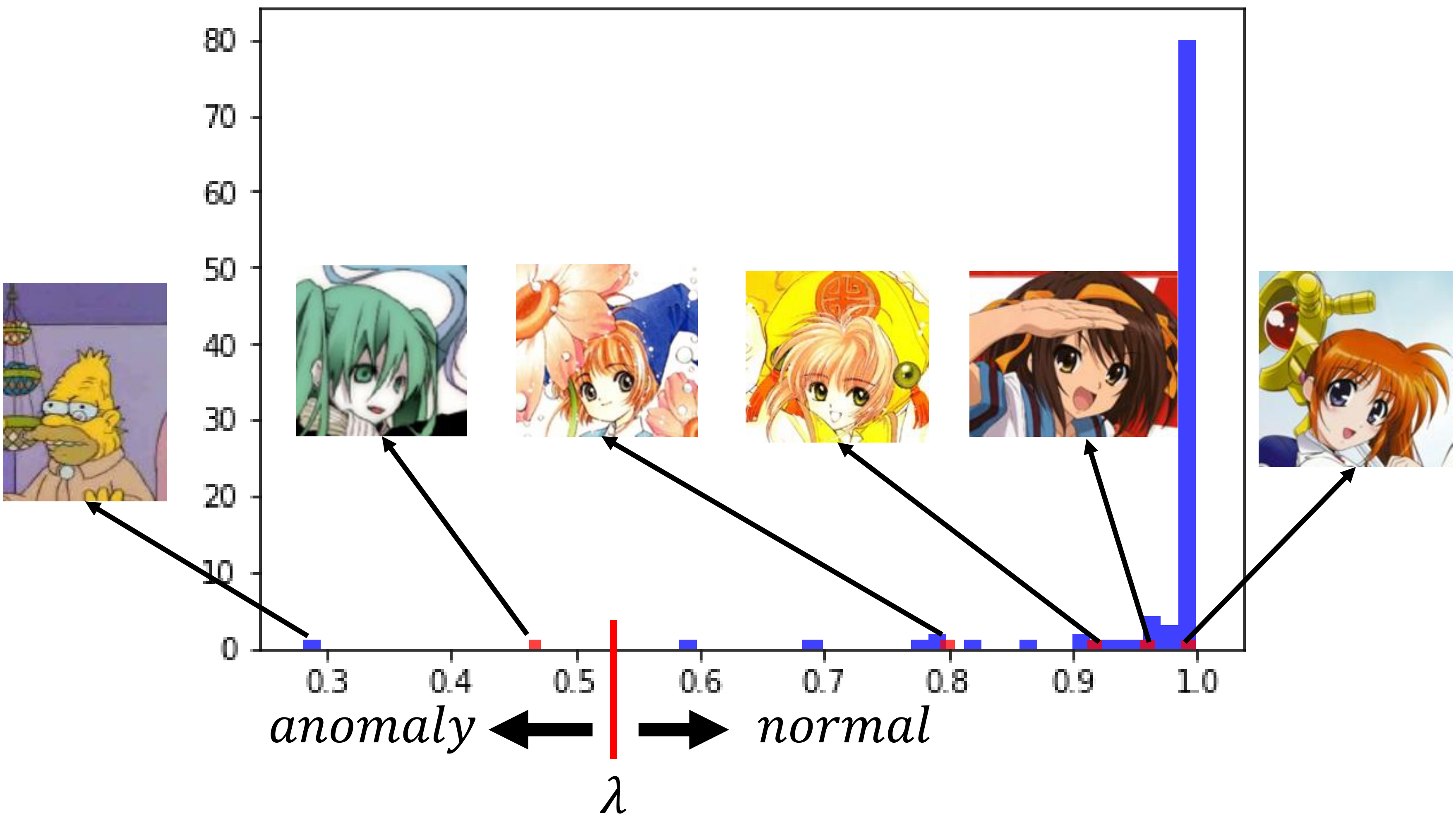


	Anomaly	Normal
Detected	1	$\boxed{1}$
Not Det	$\boxed{4}$	99

False alarm

missing

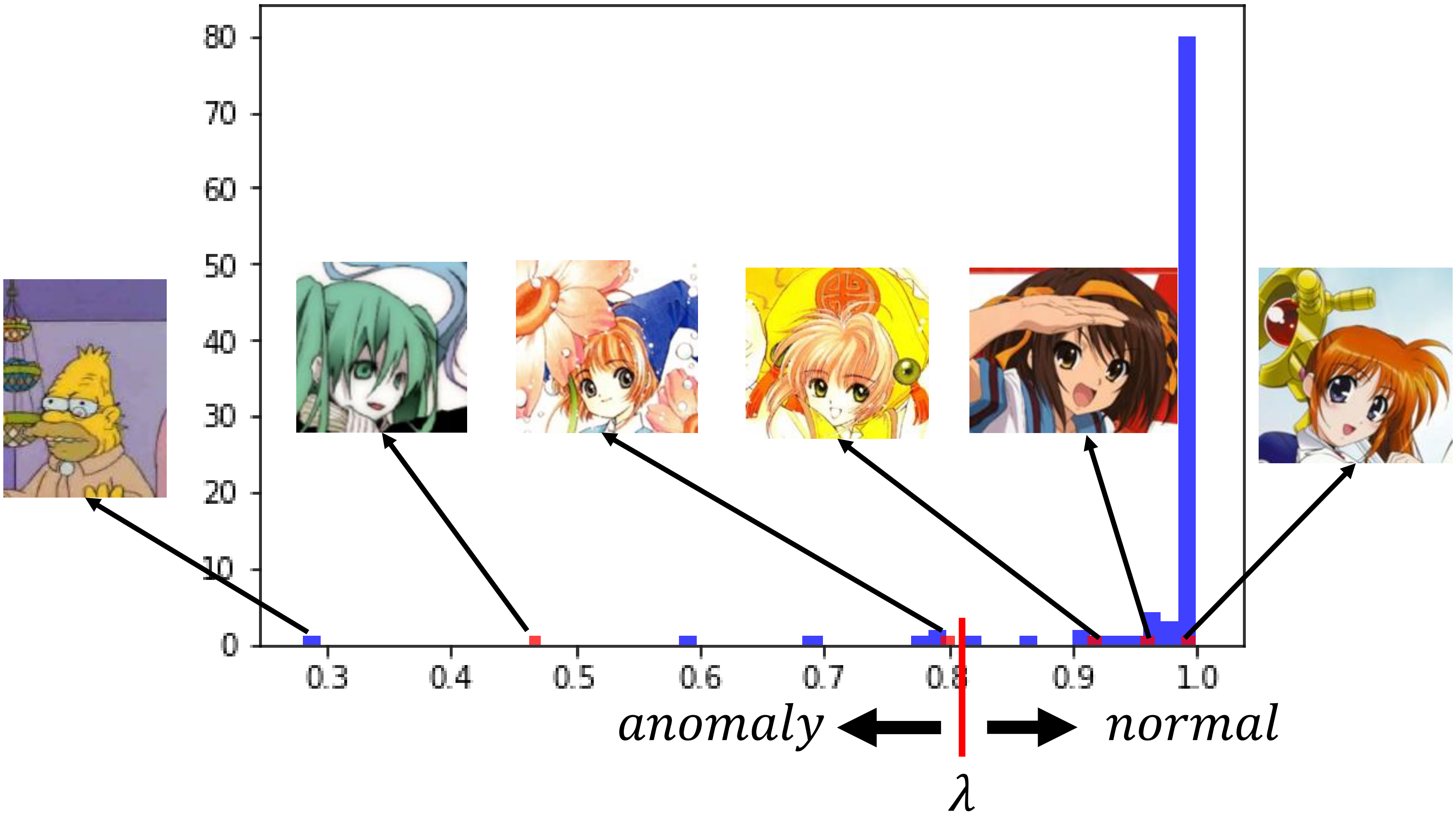
100 Simpsons, 5 anomalies



	Anomaly	Normal
Detected	1	1
Not Det	4	99

	Anomaly	Normal
Detected	2	6
Not Det	3	94

100 Simpsons, 5 anomalies



	Anomaly	Normal
Detected	1	1
Not Det	4	99

Cost = 104 勝

Cost = 401

	Anomaly	Normal
Detected	2	6
Not Det	3	94

Cost = 603

Cost = 306 勝

Cost	Anomaly	Normal
Detected	0	100
Not Det	1	0

Cost Table A

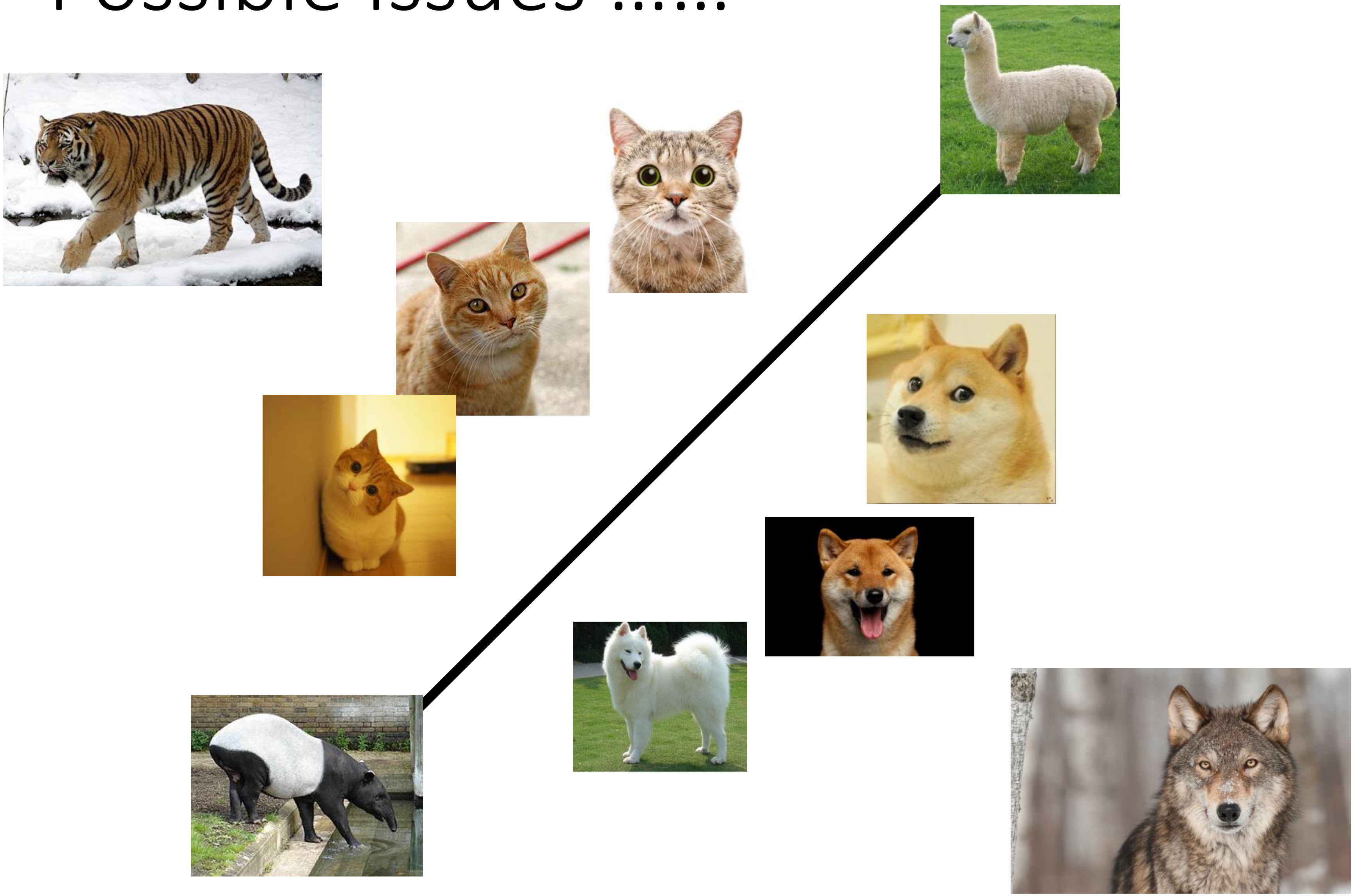
Cost	Anomaly	Normal
Detected	0	1
Not Det	100	0

Cost Table B

Some evaluation metrics consider the ranking

For example, Area under ROC curve

Possible Issues ……



Possible Issues ……

柯阿三 0.34

宅神 0.82

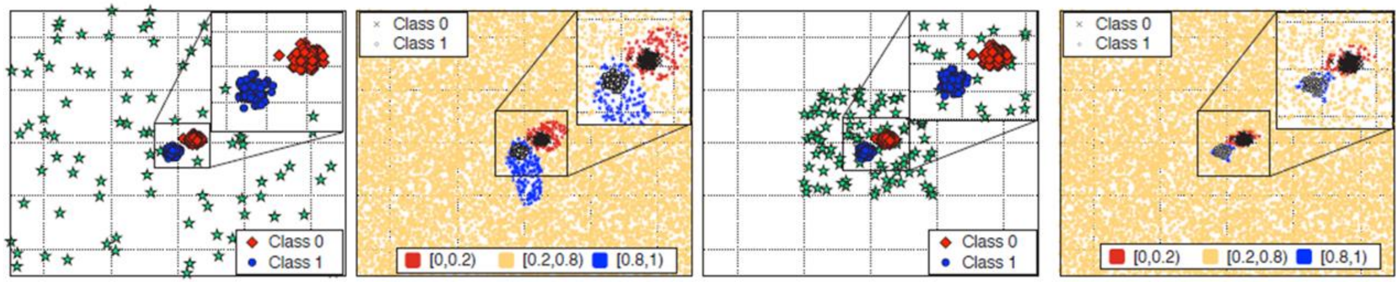
麗莎 1.00

柯阿三 0.63

麗莎 0.88

To Learn More ……

- Learn a classifier giving low confidence score to anomaly



Kimin Lee, Honglak Lee, Kibok Lee, Jinwoo Shin, Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples, ICLR 2018

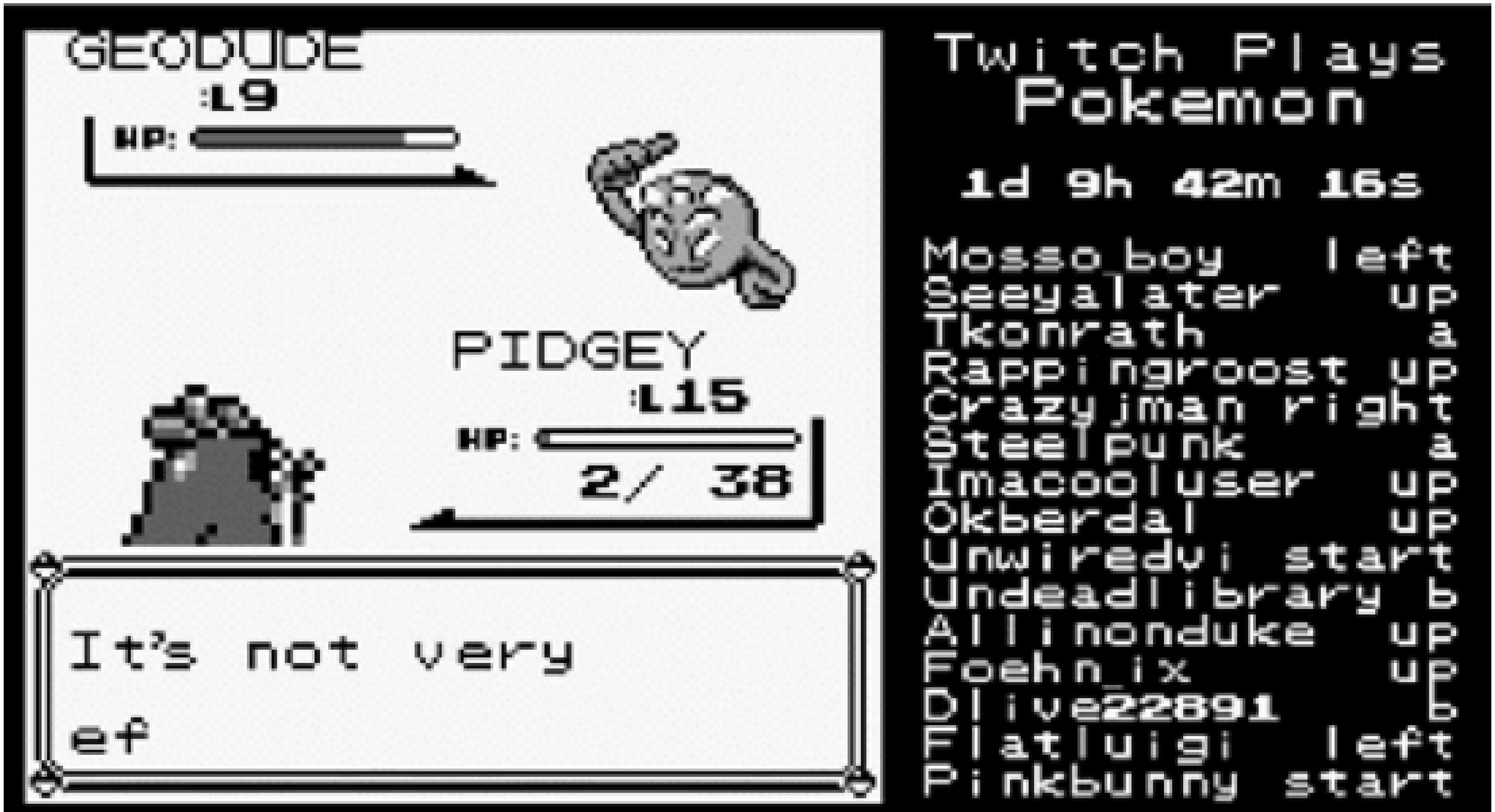
- How can you obtain anomaly?

Generating by Generative Models?

Mark Kliger, Shachar Fleishman, Novelty Detection with GAN, arXiv, 2018

Case 2: Without Labels

Twitch Plays Pokémon



Problem Formulation

- Given a set of training data $\{x^1, x^2, \cdots, x^N\}$
- We want to find a function detecting input x is *similar* to training data or not.

x

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}$$

Percent of messages that are spam (說垃圾話)

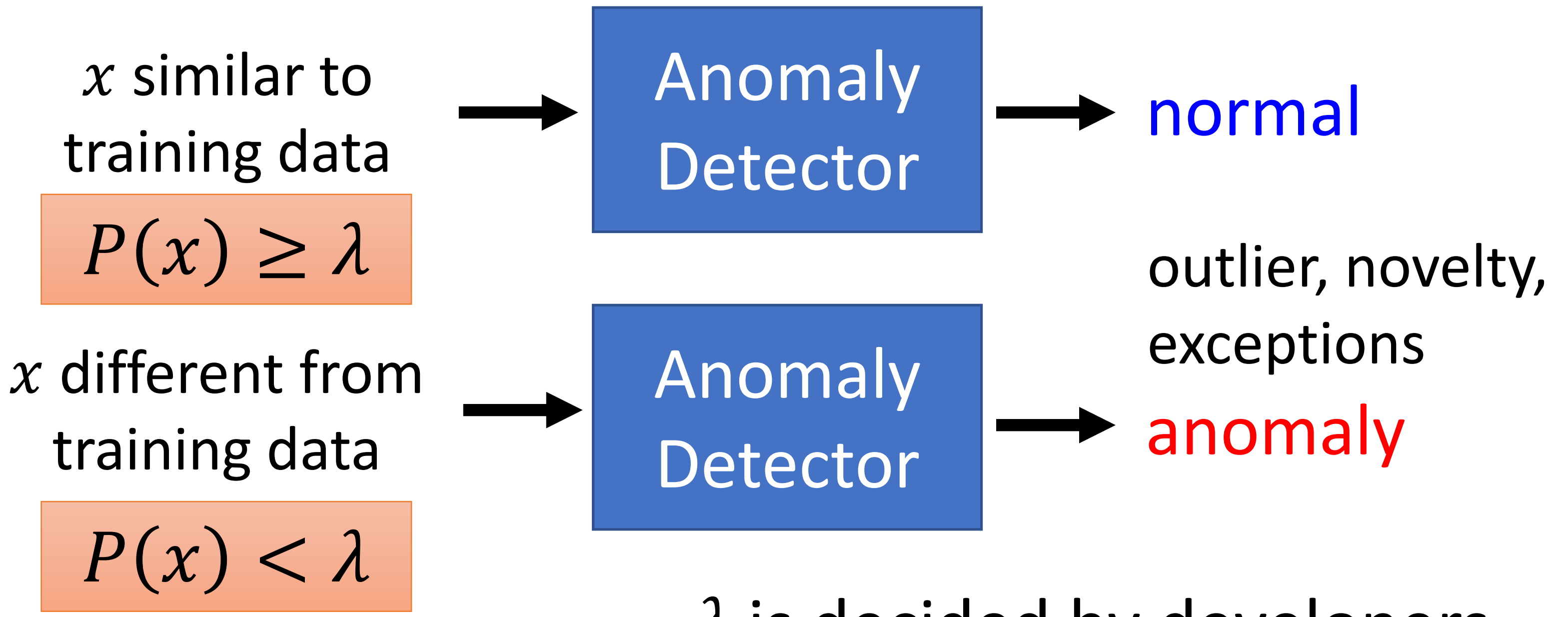
Percent of button inputs during anarchy mode (無政府狀態發言)

https://github.com/ahaque/twitch-troll-detection (Albert Haque)

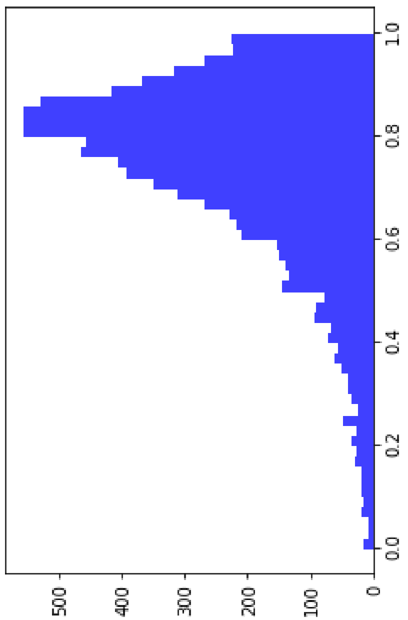
Problem Formulation

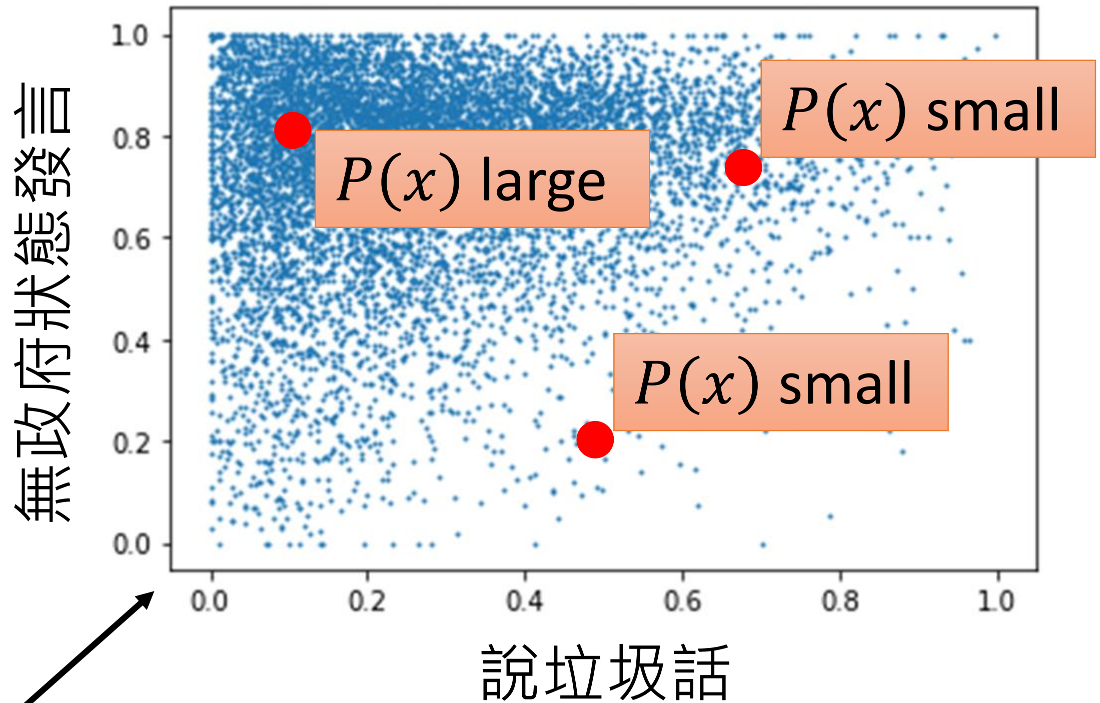
Generated from $P(x)$

- Given a set of training data $\{x^1, x^2, \cdots, x^N\}$
- We want to find a function detecting input x is *similar* to training data or not.

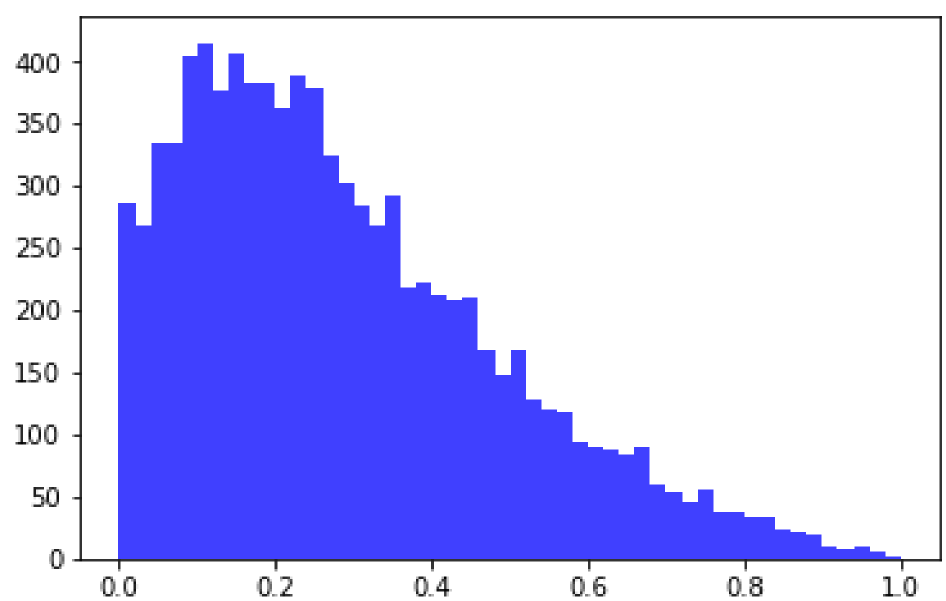


λ is decided by developers





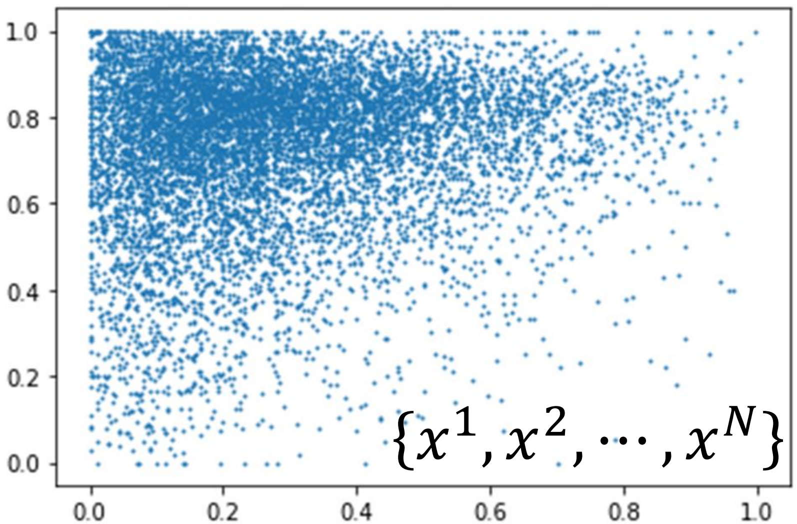
$$\{x^1, x^2, \cdots, x^N\}$$



$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Maximum Likelihood

- Assuming the data points is sampled from a probability density function $f_\theta(x)$
  - θ determines the shape of $f_\theta(x)$
  - θ is unknown, to be found from data



$$L(\theta) = f_\theta(x^1) f_\theta(x^2) \cdots f_\theta(x^N)$$

Likelihood

$$\theta^* = arg \max_\theta L(\theta)$$

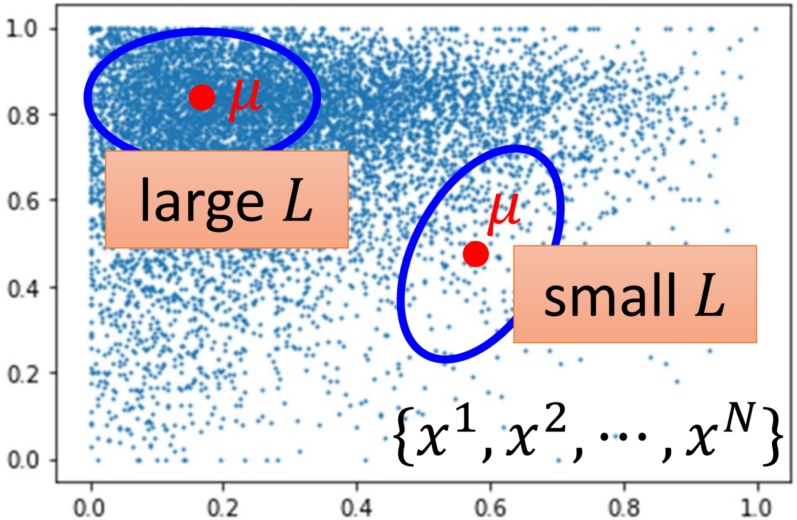
Gaussian Distribution

D is the dimension of x

$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

Input: vector x, output: probability density of sampling x

θ which determines the shape of the function are **mean** $\boldsymbol{\mu}$ and **covariance matrix** $\boldsymbol{\Sigma}$



$$L(\theta) = f_\theta(x^1) f_\theta(x^2) \cdots f_\theta(x^N)$$

$$L(\mu,\Sigma) = f_{\mu,\Sigma}(x^1) f_{\mu,\Sigma}(x^2) \cdots f_{\mu,\Sigma}(x^N)$$

$$\theta^* = arg\max_\theta L(\theta)$$

$$\mu^*, \Sigma^* = arg\max_{\mu,\Sigma} L(\mu,\Sigma)$$

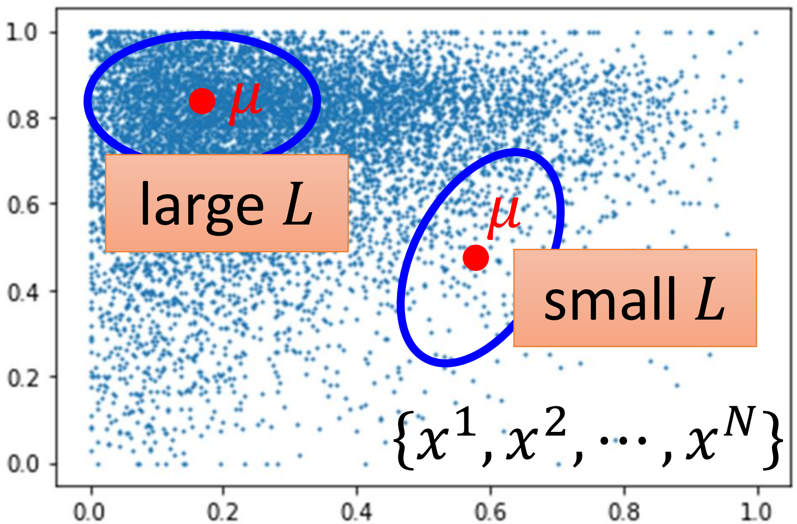
How about $f_\theta(x)$ is from a network, and θ is network parameters? (out of the scope)

Gaussian Distribution

$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

Input: vector x, output: probability of sampling x

θ which determines the shape of the function are **mean $\boldsymbol{\mu}$** and **covariance matrix $\boldsymbol{\Sigma}$**



$$L(\theta) = f_\theta(x^1)f_\theta(x^2)\cdots f_\theta(x^N)$$

$$L(\mu,\Sigma) = f_{\mu,\Sigma}(x^1)f_{\mu,\Sigma}(x^2)\cdots f_{\mu,\Sigma}(x^N)$$

$$\theta^* = arg\max_{\theta} L(\theta)$$

$$\mu^*, \Sigma^* = arg\max_{\mu,\Sigma} L(\mu,\Sigma)$$

$$\mu^* = \frac{1}{N}\sum_{n=1}^{N} x^n = \begin{bmatrix} 0.29 \\ 0.73 \end{bmatrix}$$

$$\Sigma^* = \frac{1}{N}\sum_{n=1}^{N}(x-\mu^*)(x-\mu^*)^T = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.03 \end{bmatrix}$$

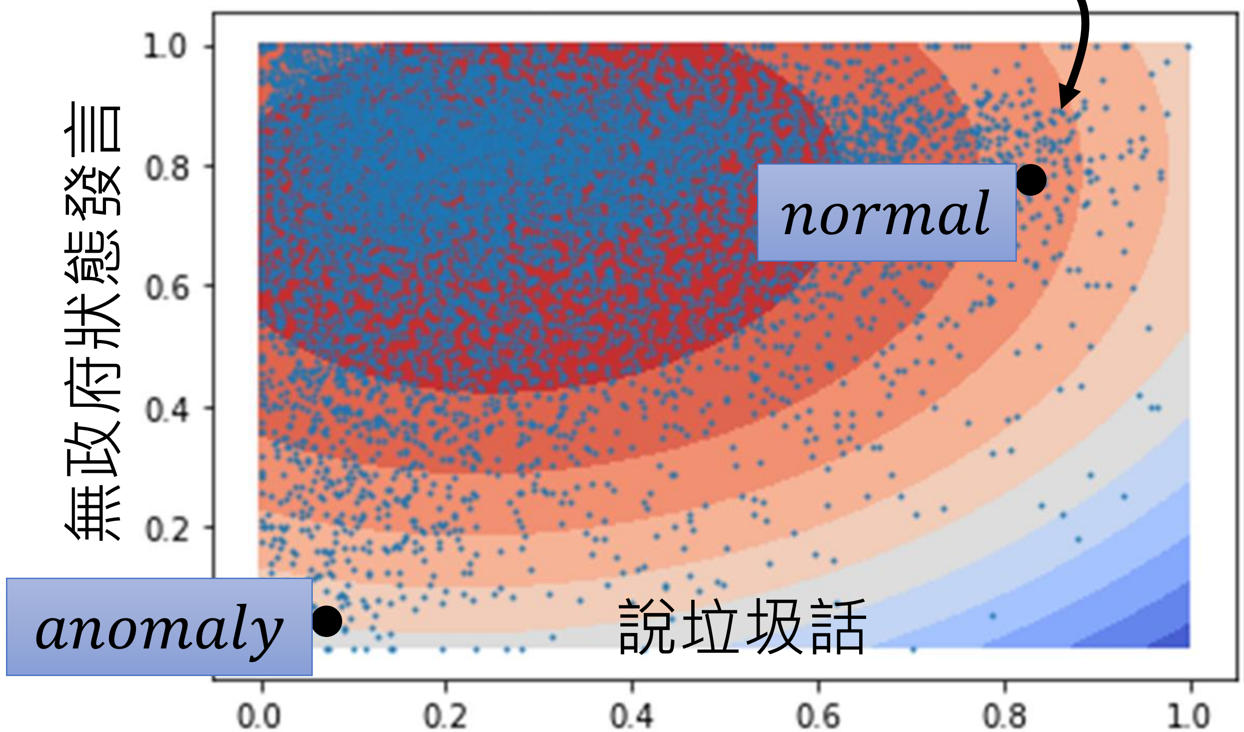
$$f_{\mu^*,\Sigma^*}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma^*|^{1/2}} exp\left\{-\frac{1}{2}(x-\mu^*)^T \Sigma^{*-1}(x-\mu^*)\right\}$$

$$\mu^* = \begin{bmatrix} 0.29 \\ 0.73 \end{bmatrix} \quad \Sigma^* = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.03 \end{bmatrix}$$

$$\boxed{f(x) = \begin{cases} normal, & f_{\mu^*,\Sigma^*}(x) > \lambda \\ anomaly, & f_{\mu^*,\Sigma^*}(x) \le \lambda \end{cases}}$$

λ is a contour line

The colors represents the value of $f_{\mu^*,\Sigma^*}(x)$



$$f(x) = \begin{cases} normal, f_{\mu^*,\Sigma^*}(x) > \lambda \\ anomaly, f_{\mu^*,\Sigma^*}(x) \le \lambda \end{cases}$$

More Features

x_1: Percent of messages that are spam (說垃圾話)

x_2: Percent of button inputs during anarchy mode (無政府狀態發言)

x_3: Percent of button inputs that are START (按 START 鍵)

x_4: Percent of button inputs that are in the top 1 group (跟大家一樣)

x_5: Percent of button inputs that are in the bottom 1 group (唱反調)

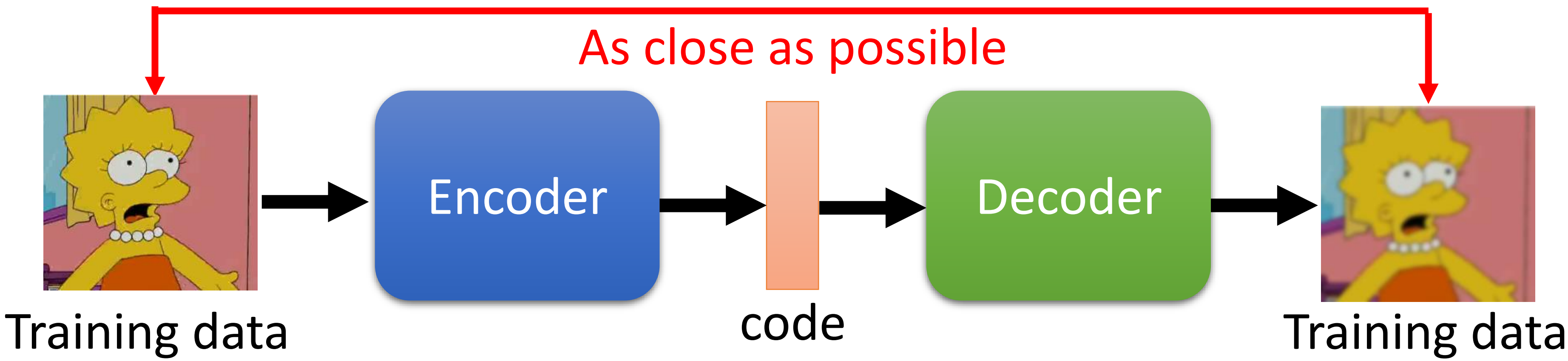
$$\begin{bmatrix} 0.1 \\ 0.9 \\ 0.1 \\ 1.0 \\ 0.0 \end{bmatrix} \rightarrow \boxed{log f_{\mu^*,\Sigma^*}(x)} \rightarrow -16$$

$$\begin{bmatrix} 0.1 \\ 0.9 \\ 0.1 \\ 0.0 \\ 0.3 \end{bmatrix} \rightarrow \boxed{log f_{\mu^*,\Sigma^*}(x)} \rightarrow -22$$

$$\begin{bmatrix} 0.1 \\ 0.9 \\ 0.1 \\ 0.7 \\ 0.0 \end{bmatrix} \rightarrow \boxed{log f_{\mu^*,\Sigma^*}(x)} \rightarrow -2$$

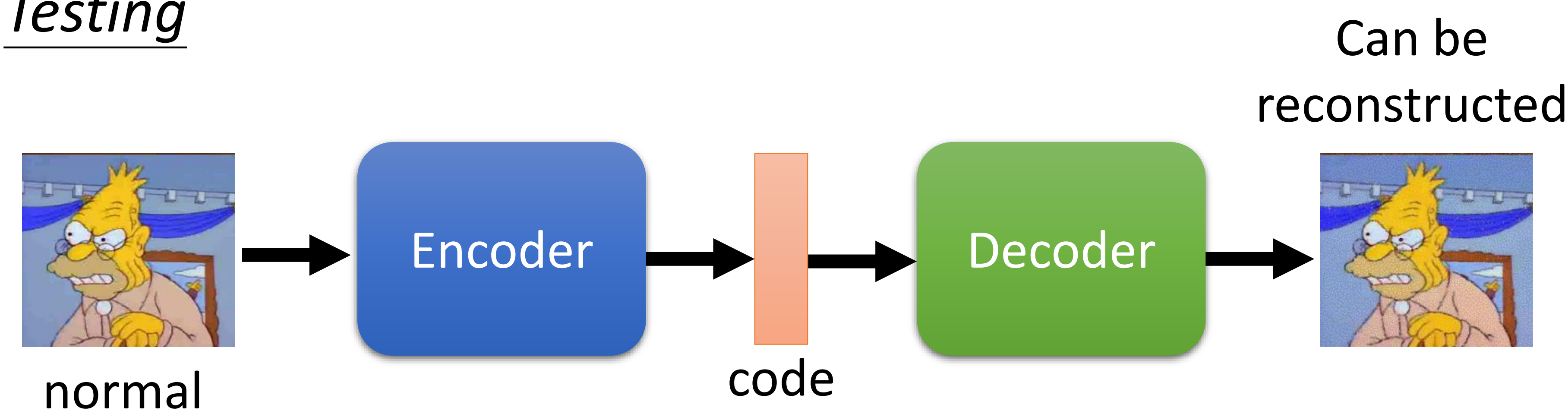
Outlook: Auto-encoder

Training



Using training data to learn an ***autoencoder***

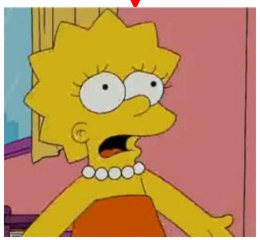
Testing



Outlook: Auto-encoder

Training

As close as possible



Training data

Encoder → code → Decoder

Training data

Using training data to learn an ***autoencoder***

Testing

Large reconstruction loss → anomaly

anomaly

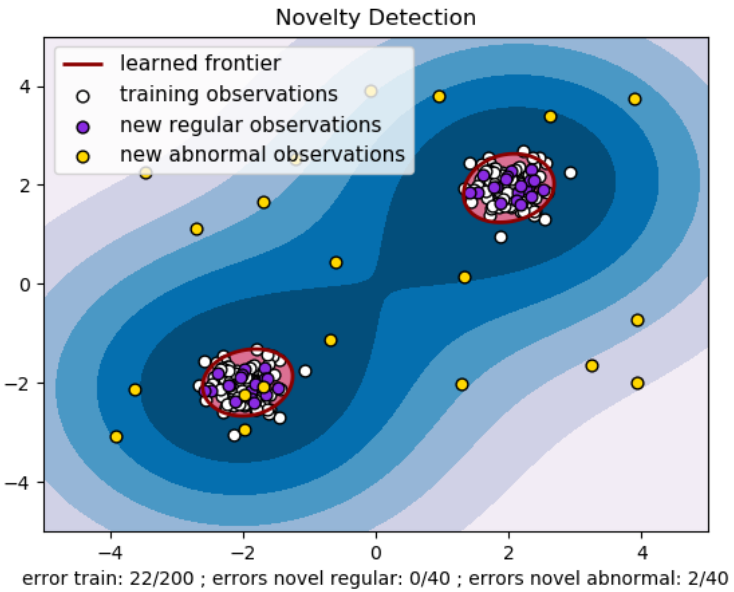
Encoder → code → Decoder

cannot be reconstructed

More …

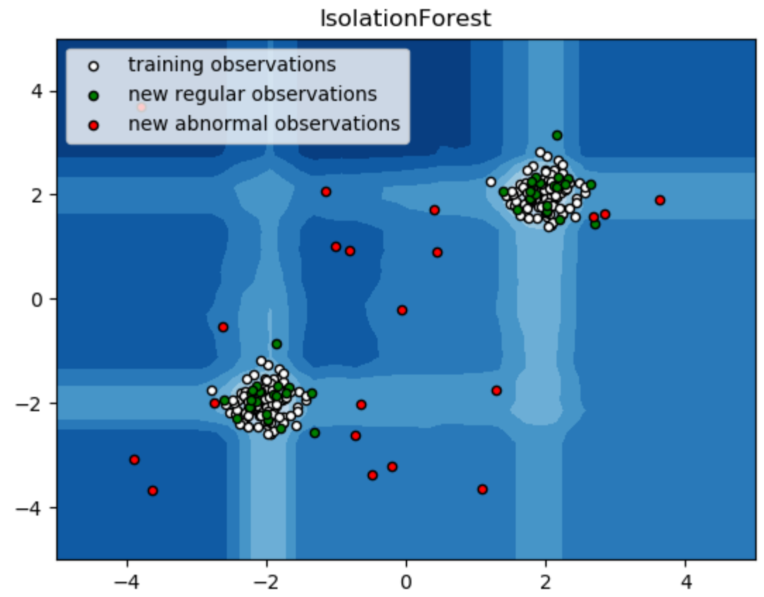
Source of images: https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection

One-class SVM



Ref: https://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection.pdf

Isolated Forest



Ref: https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf

https://www.youtube.com/watch?v=l_VsevrFHLc

Concluding Remarks

Training data:
$\{x^1, x^2, \cdots, x^N\}$

With labels:
$\{\hat{y}^1, \hat{y}^2, \cdots, \hat{y}^N\}$ → Classifier

The classifier can output "unknown" (none of the training data is labelled "unknown")

Open-set Recognition

Clean: All the training data is normal.

Polluted: A little bit of training data is anomaly.