Machine Learning Applied to Cybersecurity

Mário Antunes

July 10, 2025

Universidade de Aveiro

Context

Context

It is becoming difficult to identify Cybersecurity attacks. These attacks can originate internally due to malicious intent or negligent actions or externally by malware, target attacks, and APT (Advanced Persistent Threats).

But insider threats are more challenging and can cause more damage than external threats because they have already entered the network.

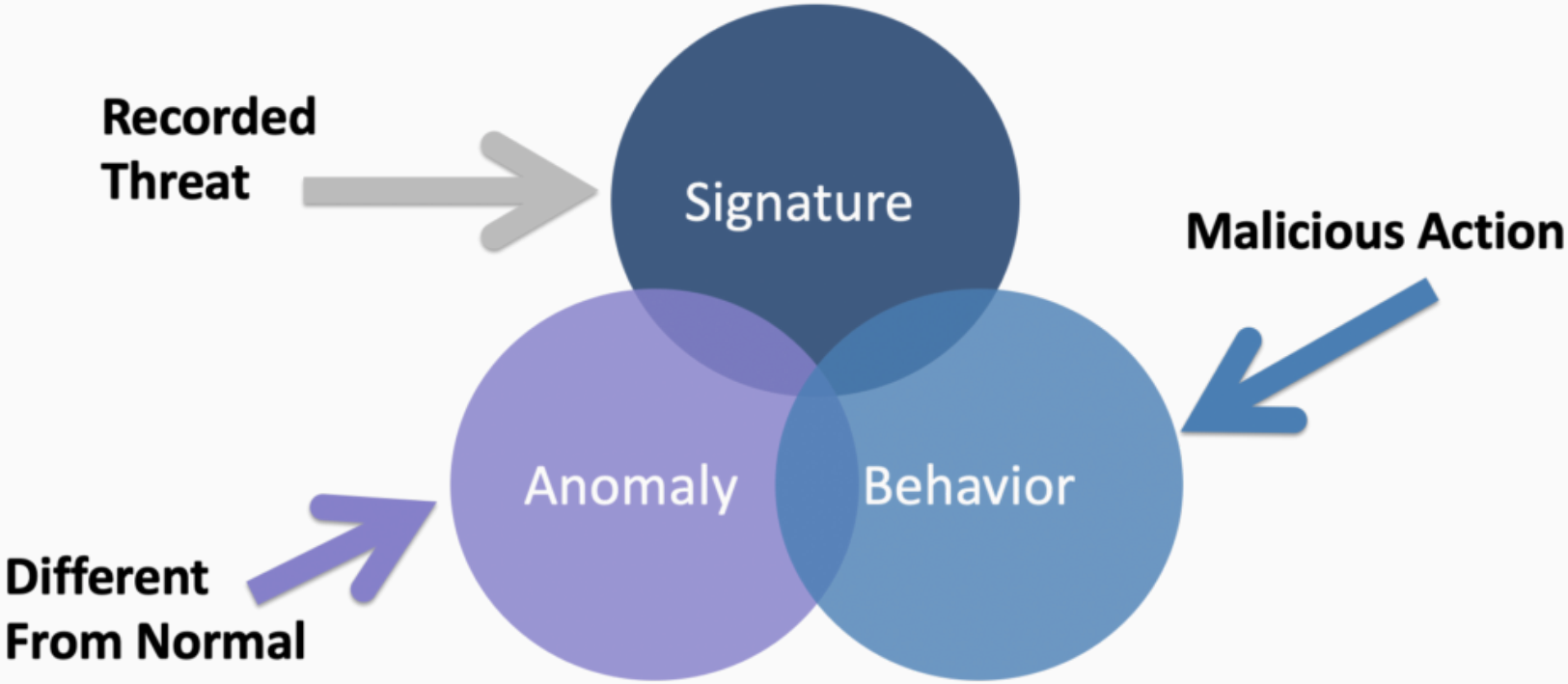
These activities present unknown threats and can steal, destroy or alter the assets.

Context #2

Earlier firewalls, web gateways, and some other intrusion prevention tools are enough to be secure, but now hackers and cyber attackers can bypass approximately all these defense systems.

Therefore with making these prevention systems strong, it is also equally essential to use detection. So that if hackers get into the network, the system should be able to detect their presence.

Context #3


Recorded
Threat
Signature
Malicious Action
Anomaly
Behavior
Different
From Normal

Context #4

Signature detection requires knowing what to look for and comparing hashes or other strings to identify a match. Signature detection is a common feature found within antivirus and IPS/IDS products.

Behavior detection looks for malicious or other known behavior characteristics and alarms the SOC when a match is made. An example is identifying port scanning or a file attempting to encrypt your hard drive, which is an indication of ransomware behavior. Antimalware and sandboxes are examples of tools that heavily leverage behavior detection capabilities.

Anomay detection it takes into consideration hot topics including big data, threat intelligence, and “zero-day” detection.

Anomalies

Anomaly Detection

Anomaly detection, also called outlier detection, is the identification of unexpected events, observations, or items that differ significantly from the norm:

- Anomalies in data occur only very rarely
- The features of data anomalies are significantly different from those of normal instances

What is an anomaly?

Generally speaking, an **anomaly** is something that differs from a norm: a deviation, an exception. In software engineering, by anomaly we understand a rare occurrence or event that doesn't fit into the pattern, and, therefore, seems suspicious. Some examples are:

- sudden burst or decrease in activity;
- error in the text logs;
- sudden rapid drop or increase in temperature.

What is an anomaly? #2

Common reasons for outliers are:

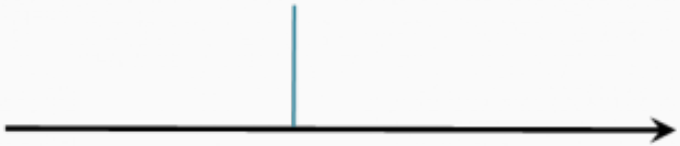
- data preprocessing errors;
- noise;
- fraud;
- attacks.

Types of Anomalies

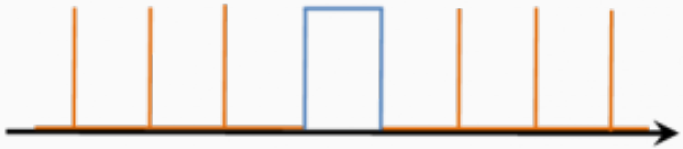
Anomalies can be broadly categorized as:

- Point anomalies: A single instance of data is anomalous if it's too far off from the rest.
- Contextual anomalies: The abnormality is context specific. This type of anomaly is common in time-series data.
- Collective anomalies: A set of data instances collectively helps in detecting anomalies.

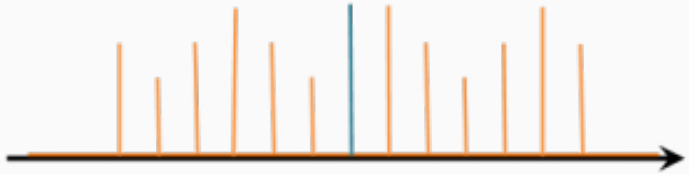
Types of Anomalies #2



(a) Point Anomaly



(b) Collective Anomaly



(c) Contextual Anomaly

Examples

Anomaly Detection - Example #1

Network anomalies: Anomalies in network behavior deviate from what is normal, standard, or expected. To detect network anomalies, network owners must have a concept of expected or normal behavior. Detection of anomalies in network behavior demands the continuous monitoring of a network for unexpected trends or events.

Anomaly Detection - Example #2

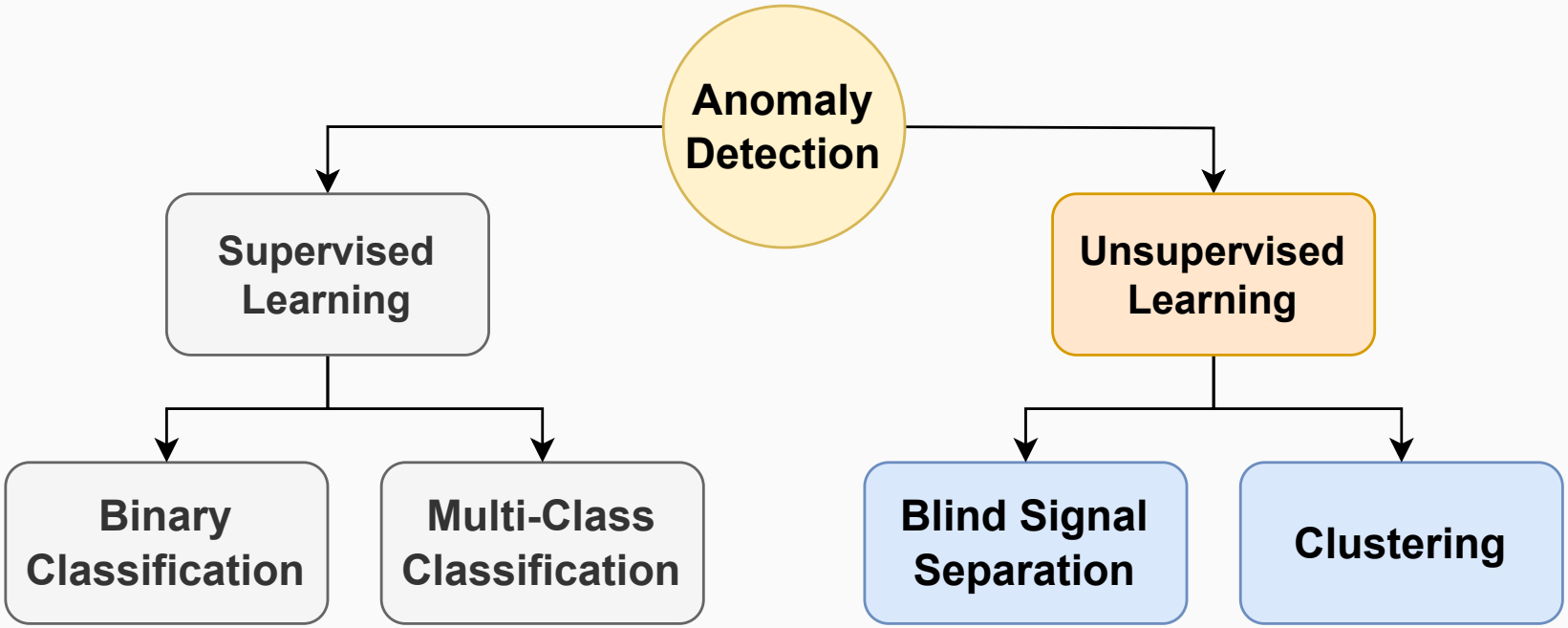
Application performance anomalies: These are simply anomalies detected by end-to-end application performance monitoring. These systems observe application function, collecting data on all problems, including supporting infrastructure and app dependencies. When anomalies are detected, rate limiting is triggered and admins are notified about the source of the issue with the problematic data.

Anomaly Detection - Example #3

Web application security anomalies: These include any other anomalous or suspicious web application behavior that might impact security such as CSS attacks or DDOS attacks.

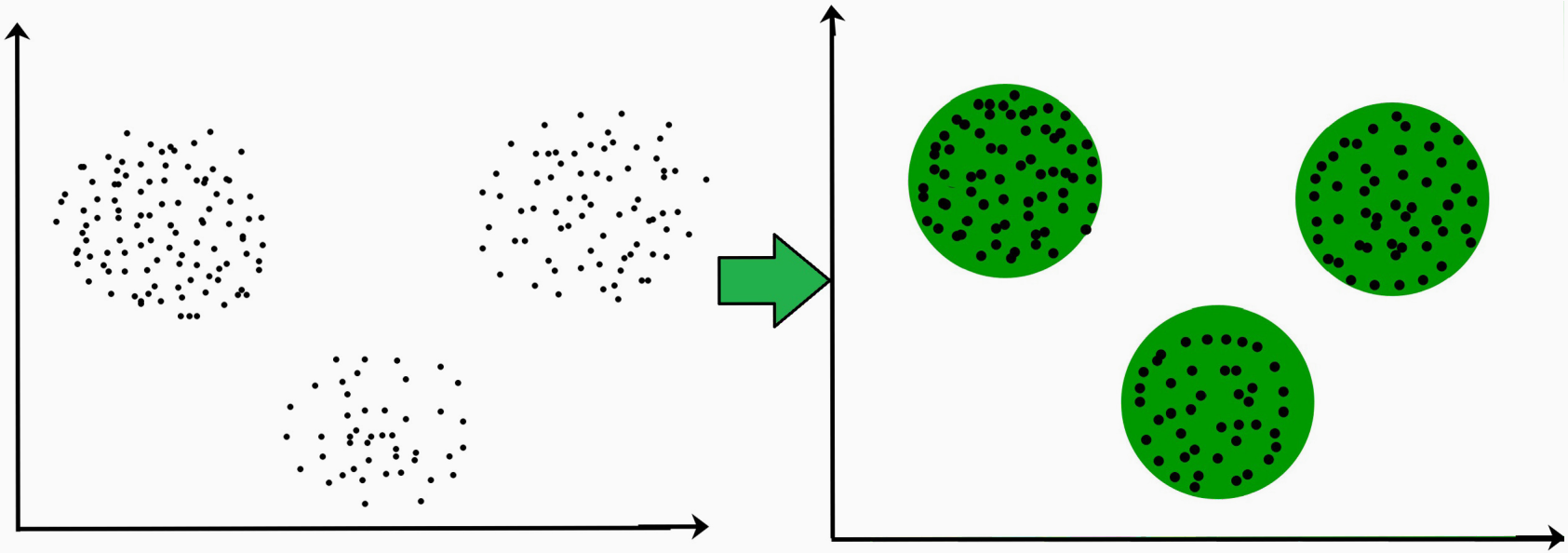
Anomaly Detection

Anomaly Detection



Clustering

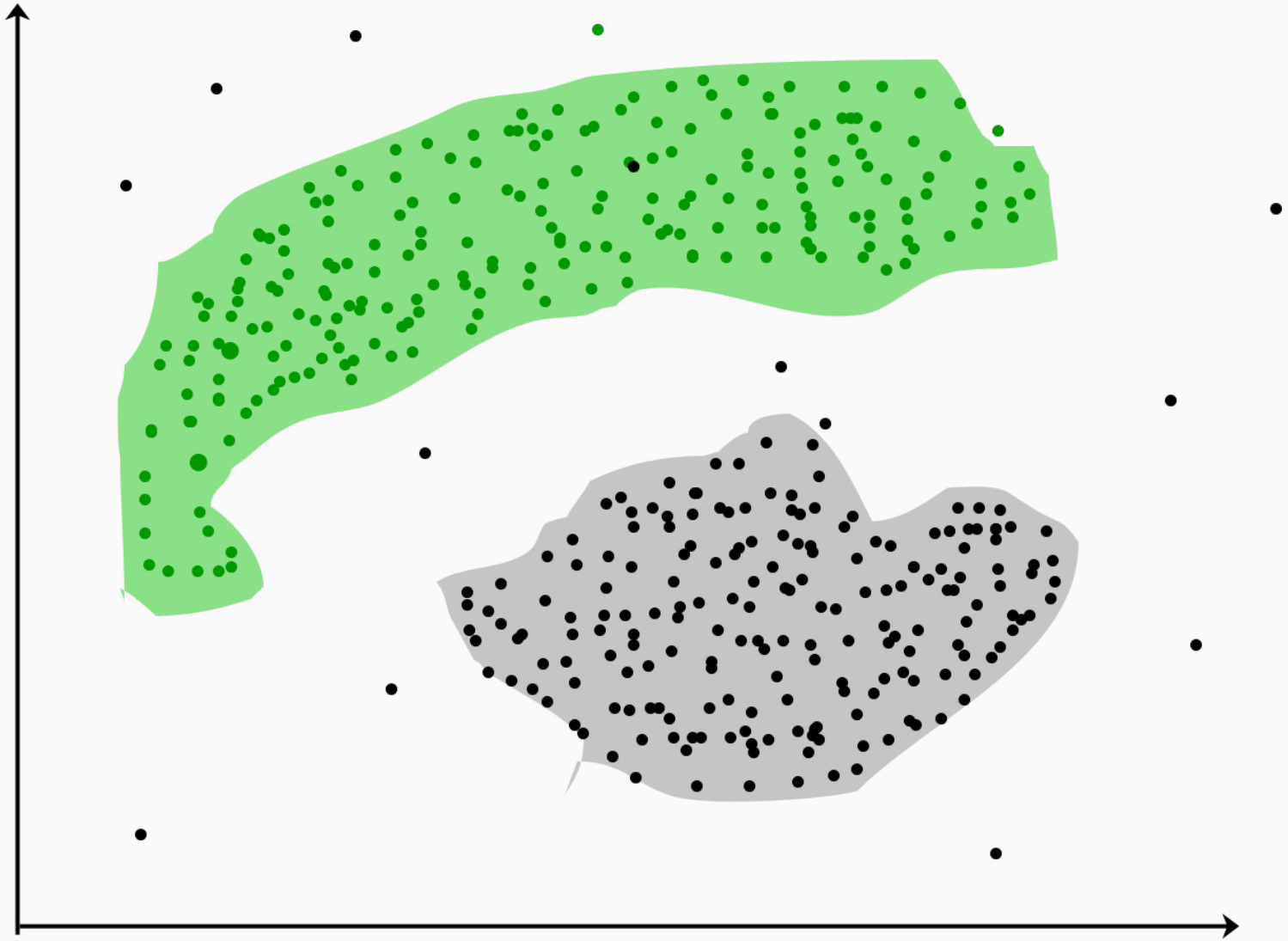
Type of **unsupervised learning method**. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.



Clustering Methods

- **Density-Based Methods:** These methods consider the clusters as the dense region having some similarities and differences from the lower dense region of the space. These methods have good accuracy and the ability to merge two clusters.
- **Hierarchical Based Methods:** The clusters formed in this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one.
- **Partitioning Methods:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter.

Clustering: Anomaly Detection



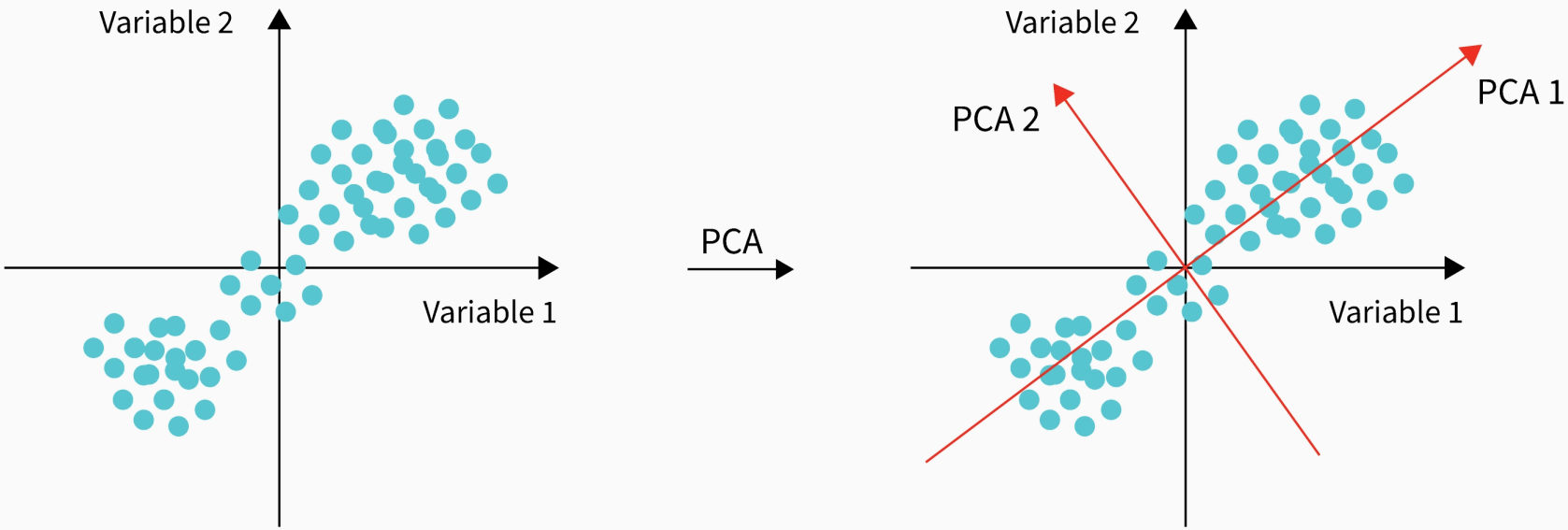
Blind Source Separation

Blind Source Separation (BSS) refers to a problem where both the sources and the mixing methodology are unknown, only mixture signals are available for further separation process.

In several situations it is desirable to recover all individual sources from the mixed signal, or at least to segregate a particular source.

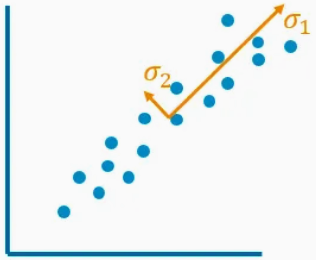
Blind Source Separation: PCA

Principal component analysis, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.

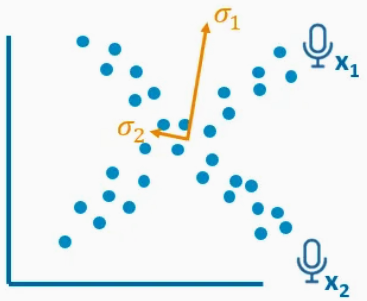


Blind Source Separation: ICA

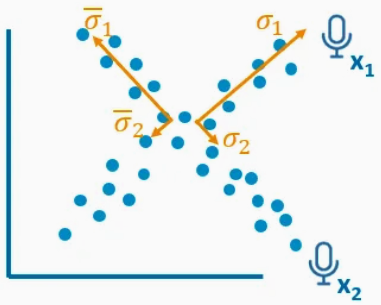
Independent Component Analysis (ICA) is a powerful technique in the field of data analysis that allows you to separate and identify the underlying independent sources in a multivariate data set.



PCA finds main directions in data:
the principal components



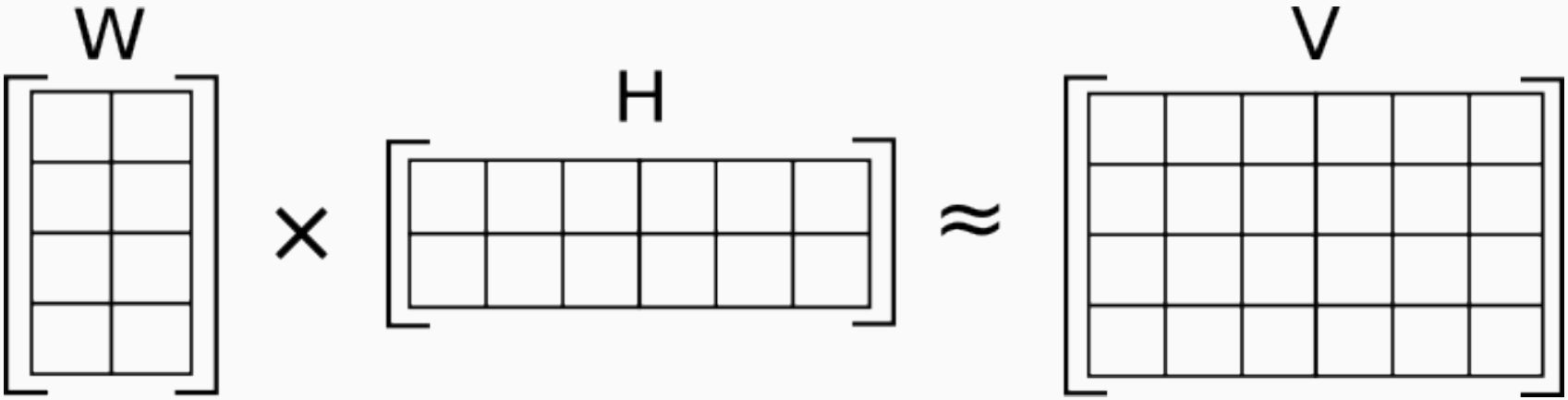
PCA fails for data sets where we have more than one principal direction



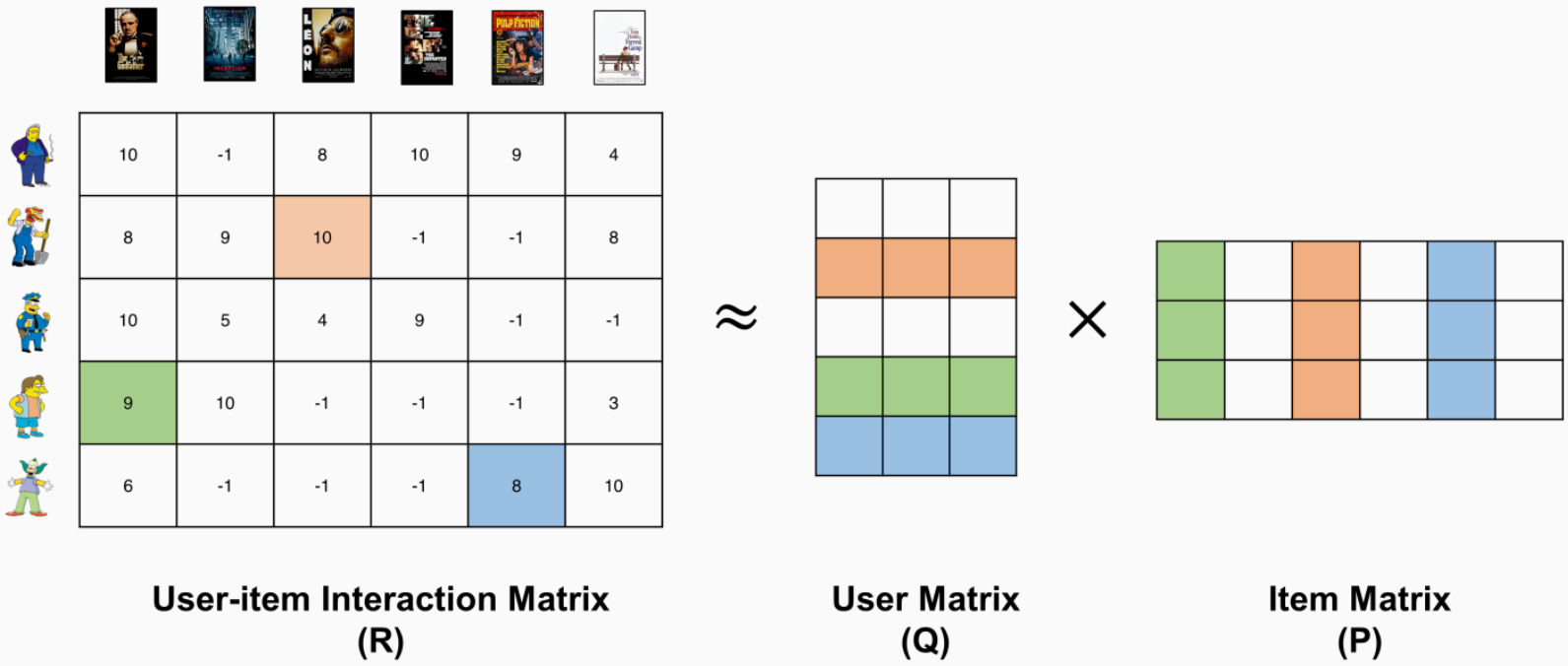
ICA solves this problem for us by focusing on independent components rather than principal components

Blind Source Separation: NNMF

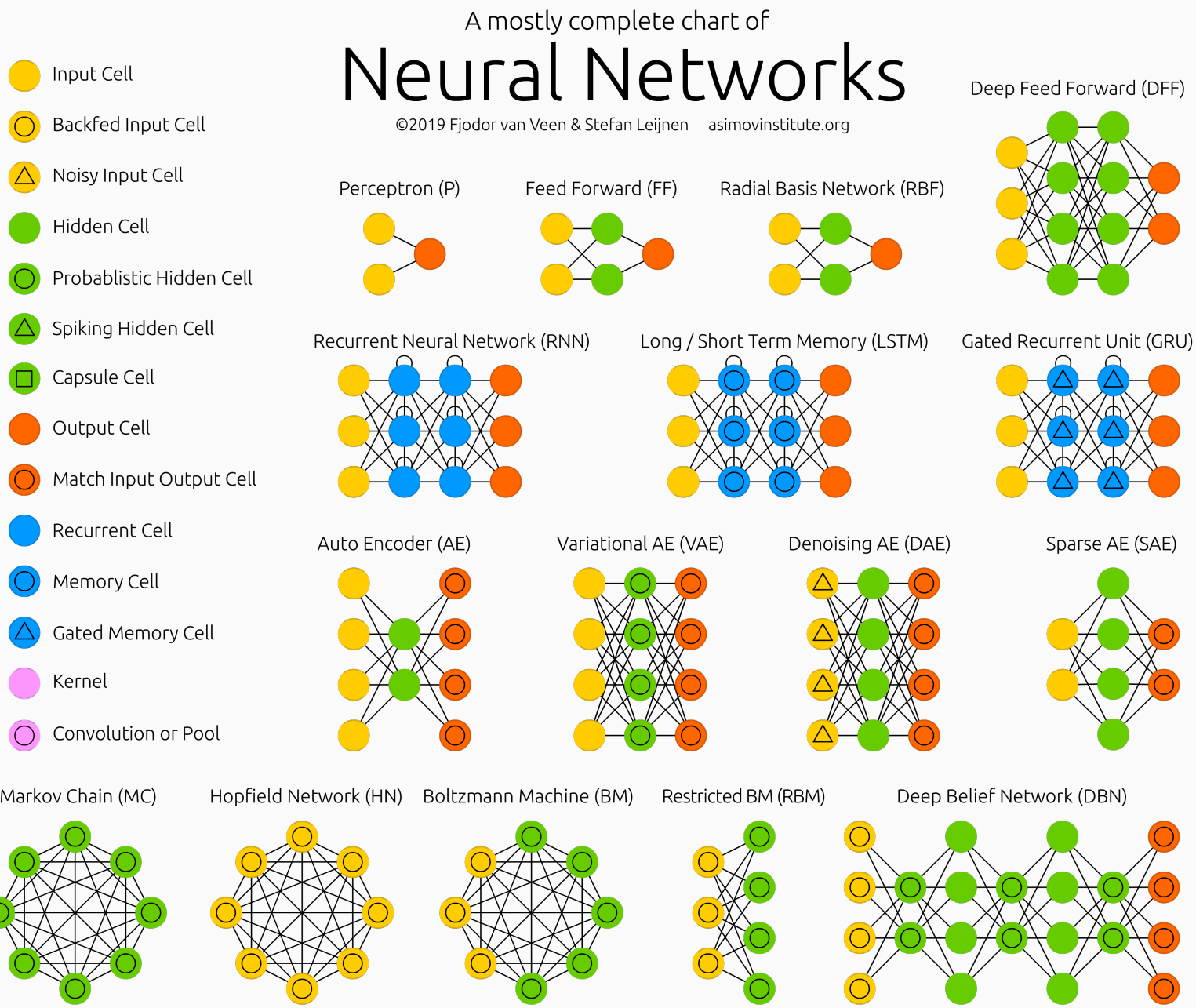
- **Non-negative matrix factorization (NNMF)** is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into two matrices W and H, with the property that all three matrices have no negative elements.
- This non-negativity makes the resulting matrices easier to inspect. Also, in applications such as processing of audio spectrograms or muscular activity, non-negativity is inherent to the data being considered.



Blind Source Separation: Anomaly Detection

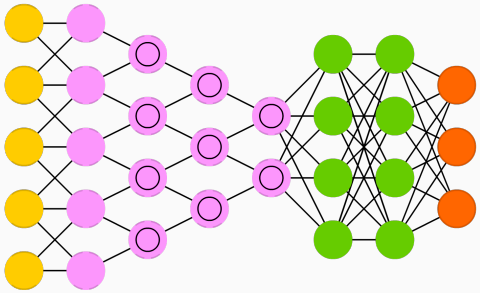


Auto Encoders

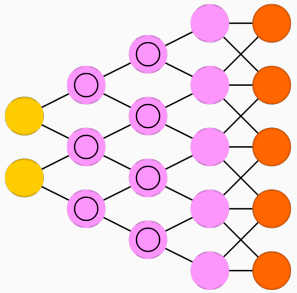

A mostly complete chart of
Neural Networks
©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org
Input Cell
Backfed Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Capsule Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Gated Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)

Neural Networks #2

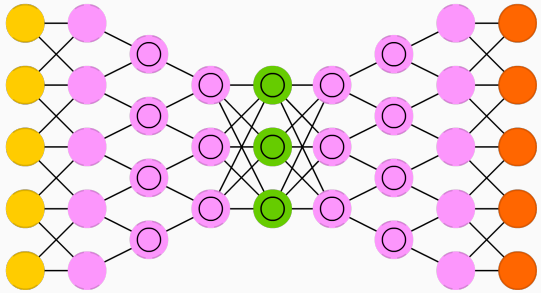
Deep Convolutional Network (DCN)



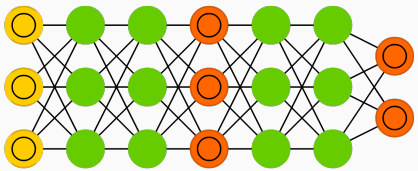
Deconvolutional Network (DN)



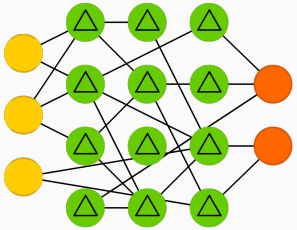
Deep Convolutional Inverse Graphics Network (DCIGN)



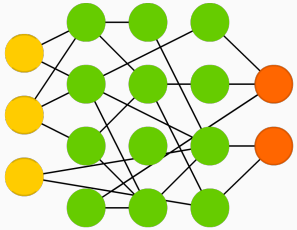
Generative Adversarial Network (GAN)



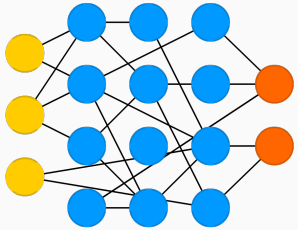
Liquid State Machine (LSM)



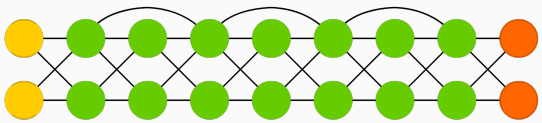
Extreme Learning Machine (ELM)



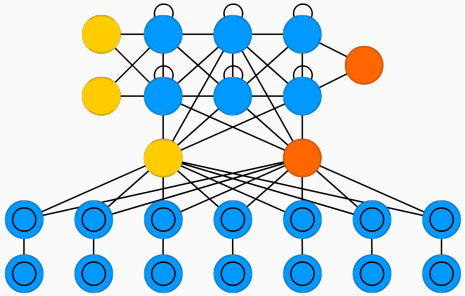
Echo State Network (ESN)



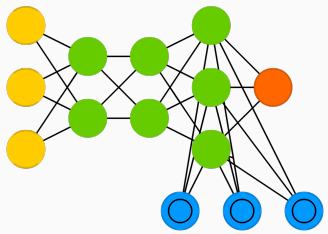
Deep Residual Network (DRN)



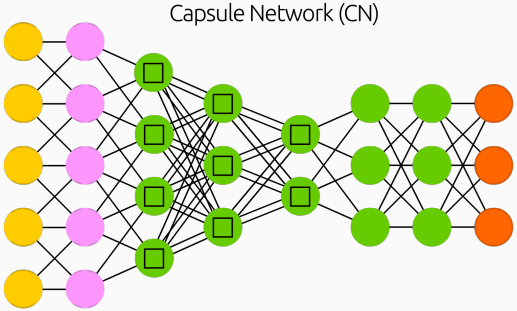
Differentiable Neural Computer (DNC)



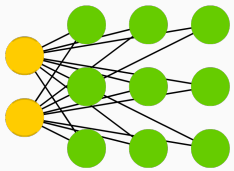
Neural Turing Machine (NTM)



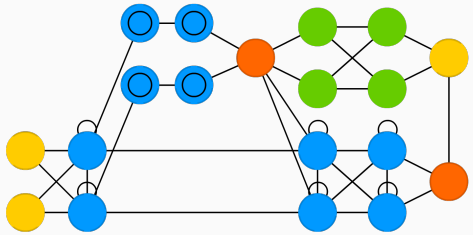
Capsule Network (CN)



Kohonen Network (KN)



Attention Network (AN)

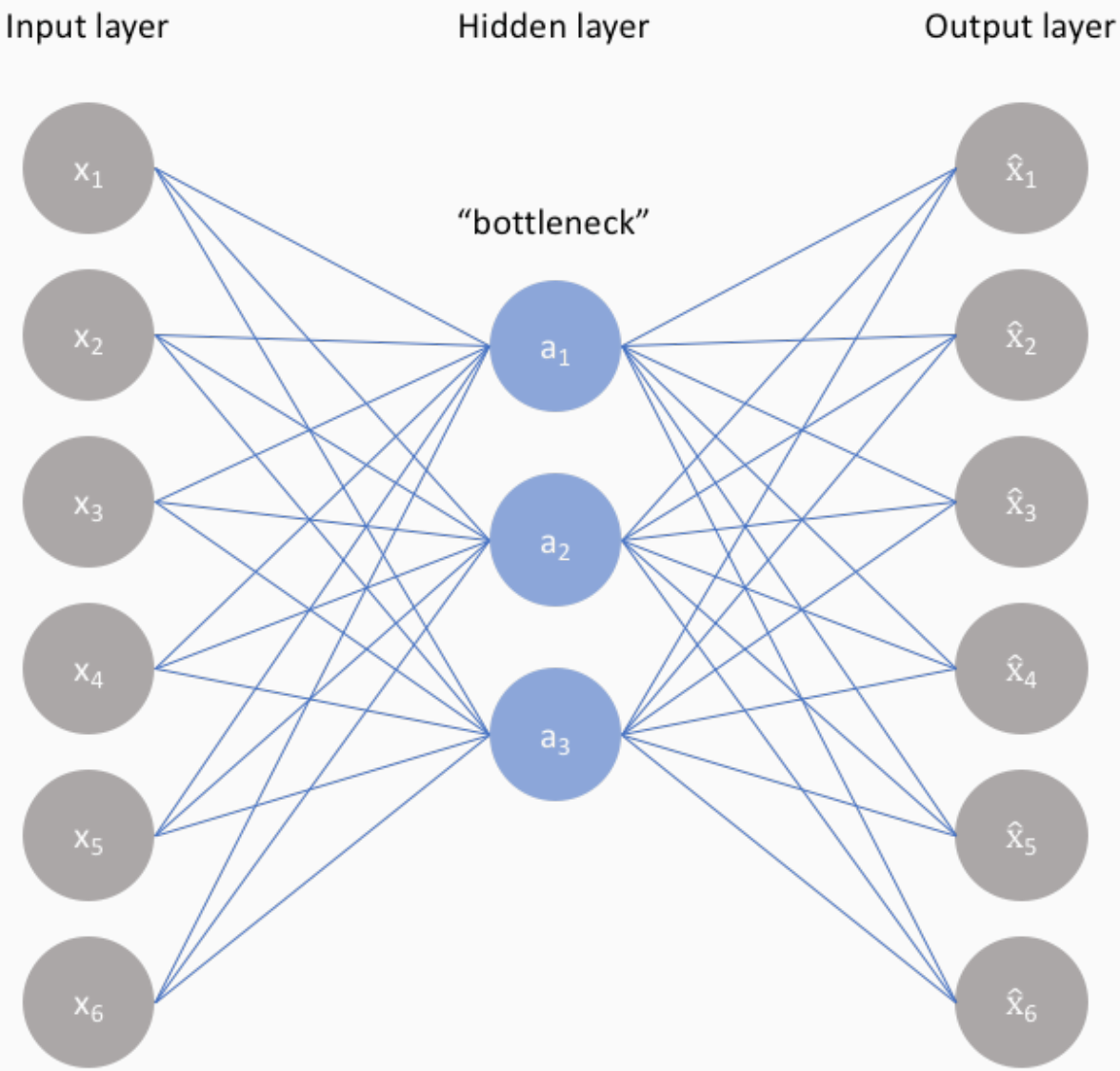


Auto Encoders

- Autoencoders are an unsupervised learning technique in which we leverage neural networks for the task of representation learning.
- Specifically, we'll design a neural network architecture such that we impose a bottleneck in the network which forces a compressed knowledge representation of the original input.

If the input features were each **independent** of one another, this compression and subsequent reconstruction would be a very **difficult task**. However, if some sort of structure exists in the data (ie. correlations between input features), this structure can be learned and consequently leveraged when forcing the input through the network's bottleneck.

Auto Encoders #2



Auto Encoders #3

- As visualized, we can take an unlabeled dataset and frame it as a supervised learning;
- This network can be trained by minimizing the reconstruction error;
- The bottleneck is a key attribute of our network design; without the presence of an information bottleneck, our network could easily learn to simply memorize the input values by passing these values along through the network.

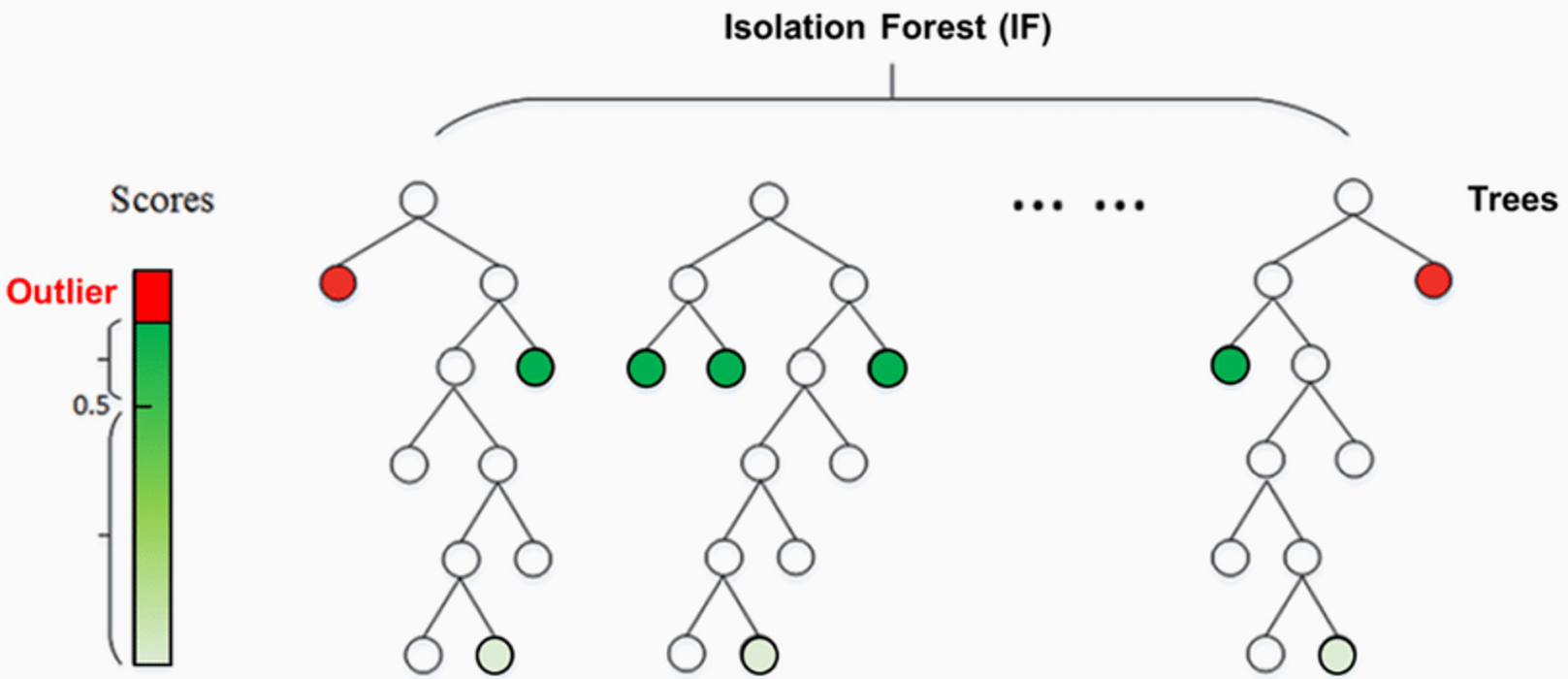
Other Methods

Anomaly Detection - IsolationForest

- IsolationForest **isolates** observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
- Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node.
- This path length, averaged over a forest of such random trees, is a measure of normality and our decision function.

Random partitioning produces noticeably shorter paths for anomalies. Hence, when a forest of random trees collectively produce shorter path lengths for particular samples, they are highly likely to be **anomalies**.

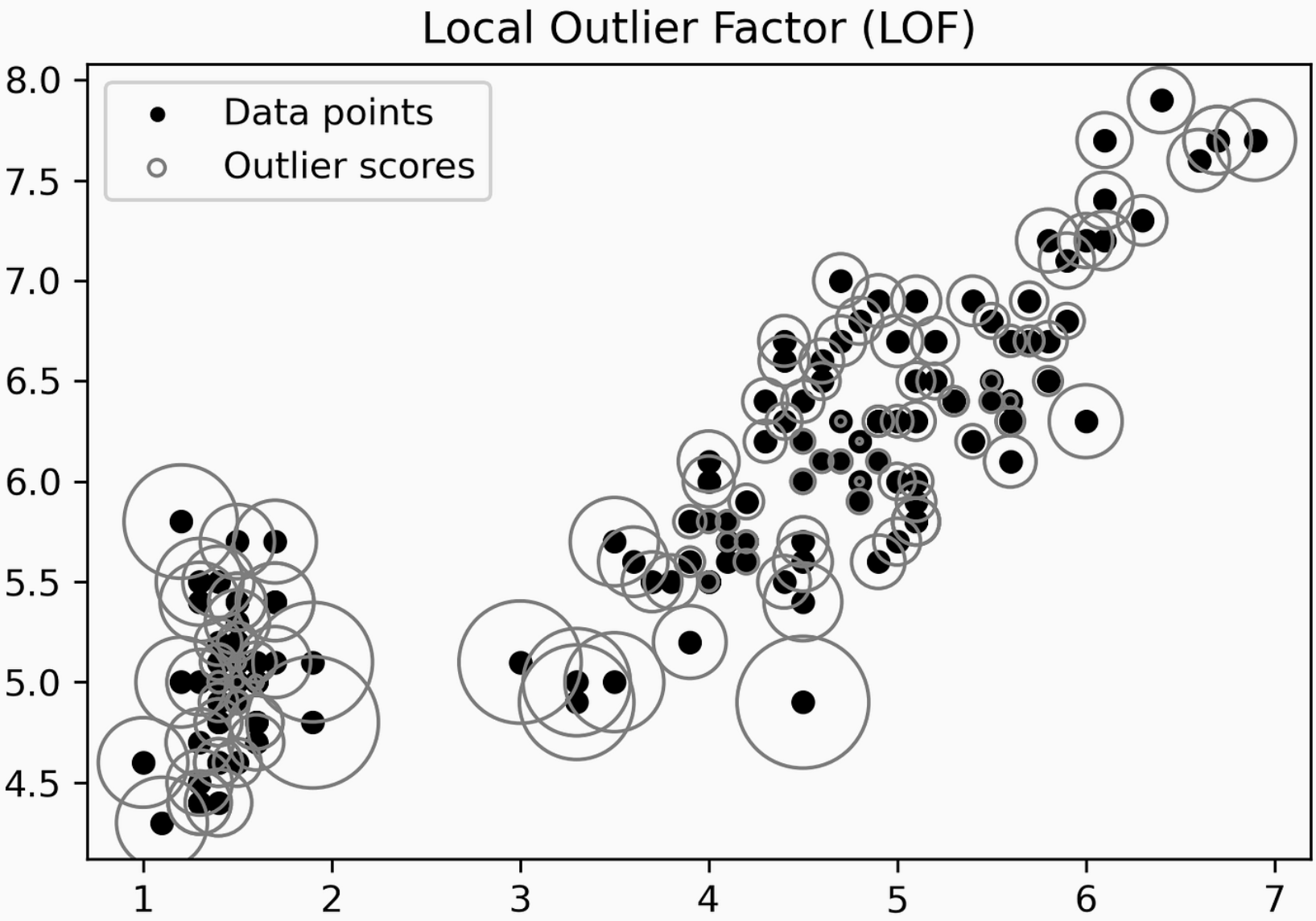
Anomaly Detection - IsolationForest #2



Anomaly Detection - Local Outlier Factor

- Local Outlier Factor (LOF) measures the local deviation of the density of a given sample with respect to its neighbors.
- It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood.
- More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers.

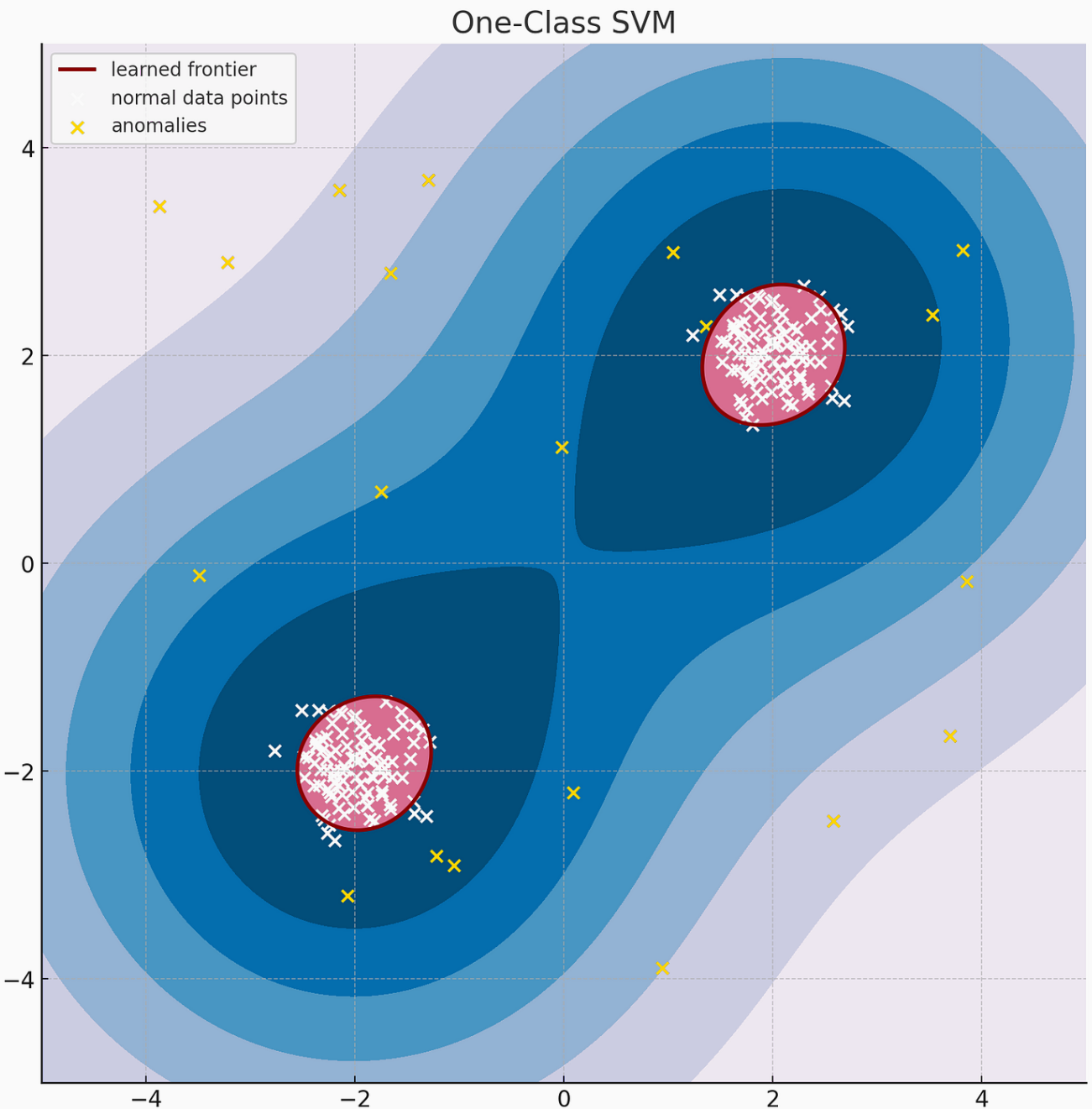
Anomaly Detection - Local Outlier Factor #2



Anomaly Detection - OneClassSVM

- Many approaches are based on the estimation of the density of probability for the normal data. Anomlies corresponds to those samples where the density of probability is “very low”.
- Now, SVMs are max-margin methods, i.e. they do not model a probability distribution. Here the idea is to find a function that is positive for regions with high density of points, and negative for small densities.
- One-Class SVM is similar, but instead of using a hyperplane to separate two classes of instances, it uses a hypersphere to encompass all of the instances. Now think of the “margin” as referring to the outside of the hypersphere – so by “the largest possible margin”, we mean “the smallest possible hypersphere”.

Anomaly Detection - OneClassSVM #2



Tips

Feature Scalling

- Feature scaling is the process of normalizing the range of features in a dataset.
- Real-world datasets often contain features that are varying in degrees of magnitude, range, and units.
- Therefore, in order for machine learning models to interpret these features on the same scale, we need to perform feature scaling.

Resources

Resources

Neural Networks Zoo

AutoEncoders

Principal components analysis