# Efficient Data Stream Mining

Talk at Complex Network, LIP6

Maroua Bahri

Inria

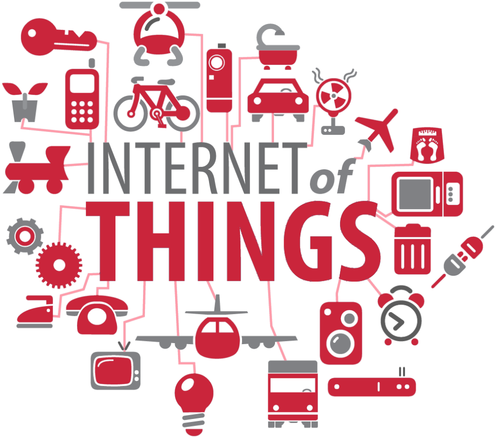
Paris, 27 June 2022

# Who am I

- **2015-2016 :** Master in Data Mining and Knowledge Management, Polytech' Nantes
- **2014-2016 :** Master in Sciences and Technologies of BI, Institut Supérieur de Gestion de Tunis
- **2017-2020 :** Ph.D. degree in Computer Science, Télécom Paris
  - Improving IoT data stream analytics using summarization techniques
  - Defended in June 2020
- **2020-2021 :** Postdoc, Télécom Paris
- **2021-Current :** Postdoc, INRIA Paris

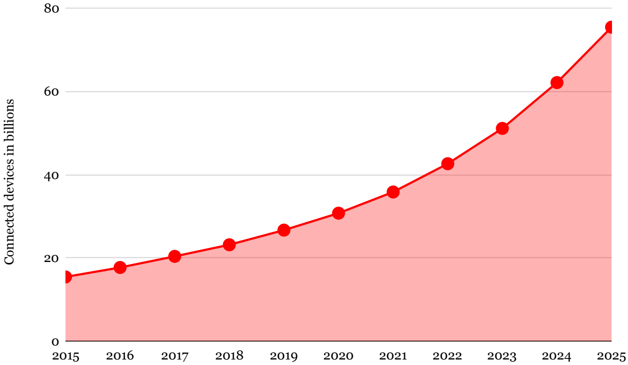

Machine Learning
supervised learning
Efficiency
prediction
classification
AutoML
IoT
big data
Data streams
dimensionality reduction
learning task
Summarization
approaches
techniques
supervised
classifiers
data processing
decision
devices
tasks
supervised learning

# Internet of Things (IoT)




- Network of connected devices

## Internet of Things (IoT)




- Statista predicts around 80 billion IoT devices by 2025

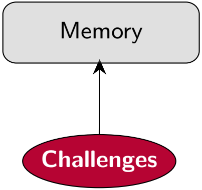
# Challenges

- Technical
  - Complex data
  - Computational resource
- Energetic
  - The electronic industry is leaving unfavourable environmental footprints
  - Reduction of energy supply
- Security
  - Ensuring security in IoT products and services
- Economic
  - Some materials are rare or becoming
- ...

# Challenges

- **Technical**
  - Complex data
  - Computational resource
- Energetic
  - The electronic industry is leaving unfavourable environmental footprints
  - Reduction of energy supply
- Security
  - Ensuring security in IoT products and services
- Economic
  - Some materials are rare or becoming
- ...

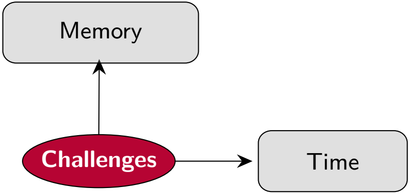
# Technical Challenges




**Memory**

- Use a limited amount of memory

# Technical Challenges



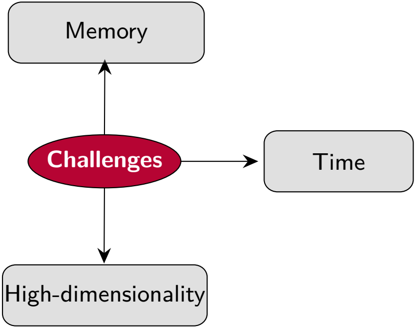

**Memory**

- Use a limited amount of memory

**Time**

- Work in a limited amount of time

# Technical Challenges




**Memory**

- Use a limited amount of memory

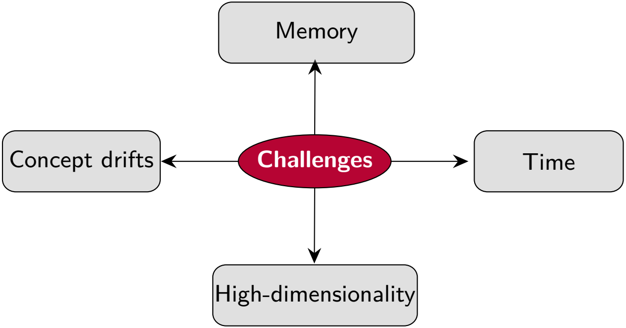
**Time**

- Work in a limited amount of time

**Dimensionality**

- Handle high-dimensional data

# Technical Challenges




**Memory**

- Use a limited amount of memory

**Time**

- Work in a limited amount of time

**Dimensionality**

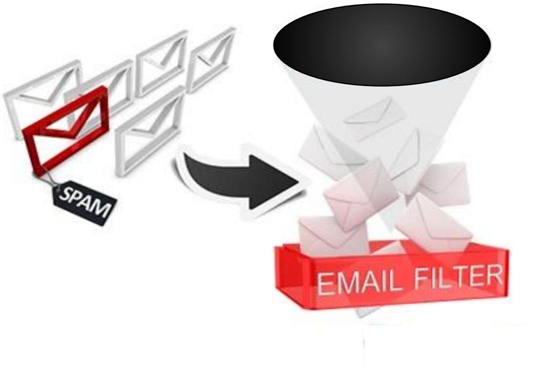
- Handle high-dimensional data

**Concept drifts**

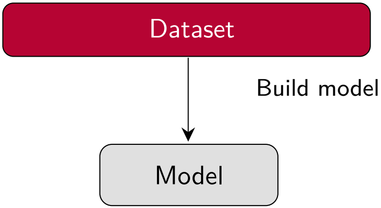
- Detect and adapt to changes

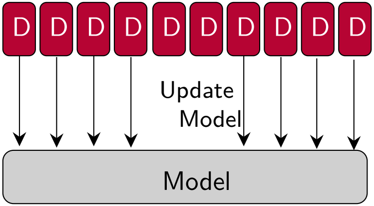
# Example : Email Filtering

# Example : Email Filtering



# Example : Email Filtering




## Batch approaches

- Finite training sets
- Static models

## Stream approaches

- Infinite training sets
- Dynamic models

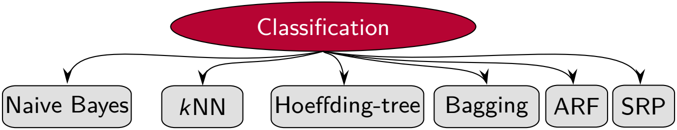
# Stream Mining

Maintain models in an online fashion

- Incorporate data on the fly
- Single pass, one instance at a time
- Once processed, it is discarded or archived
- Be ready to predict at any instance

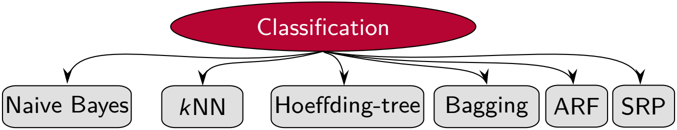
## Classification and Contributions

- Different classifiers that **continuously** operate and incorporate instances as they arrive exist [DMKD'21]



## Classification and Contributions

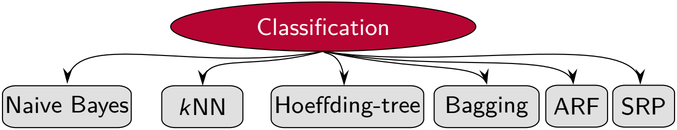
- Different classifiers that **continuously** operate and incorporate instances as they arrive exist [DMKD'21]




Accurate models but expensive, especially with high-dimensional data

# Classification and Contributions

- Different classifiers that **continuously** operate and incorporate instances as they arrive exist [DMKD'21]




Accurate models but expensive, especially with high-dimensional data

- Improve stream algorithm performance
- Guarantee a good precision
- Tradeoff between resources and accuracy

$\Rightarrow$ **Sampling, sketching, dimensionality reduction,** $\cdots$

# Compressed $k$-Nearest Neighbors [ECAI'20]

Stream $k$NN :

- Uses a sliding window as a search space
- Given an unclassified instance $X_i$ from a stream $S$ :
  - Determines the $k$NN inside the window
  - Predicts the most frequent label

# Compressed $k$-Nearest Neighbors [ECAI'20]

Stream $k$NN :

- Uses a sliding window as a search space
- Given an unclassified instance $X_i$ from a stream $S$ :
  - Determines the $k$NN inside the window
  - Predicts the most frequent label

- Inefficient at prediction time
- Memory consuming
- Inefficient with high-dimensional data

# Compressed $k$-Nearest Neighbors [ECAI'20]

Stream $k$NN :

- Uses a sliding window as a search space
- Given an unclassified instance $X_i$ from a stream $S$ :
  - Determines the $k$NN inside the window
  - Predicts the most frequent label

- Inefficient at prediction time
- Memory consuming
- Inefficient with high-dimensional data

$\Rightarrow$ **Dimensionality reduction**

# Dimensionality Reduction (DR)

The projection of high-dimensional data into a low-dimensional space by reducing the input features

*Objectif* : given an instance $X_i \in \mathbb{R}^a$, we wish to obtain $Y_i \in \mathbb{R}^m$, where $m \ll a$

- Principal Component Analysis (PCA)
- Compressed Sensing (CS)
- Hashing Trick (HT)
- ...

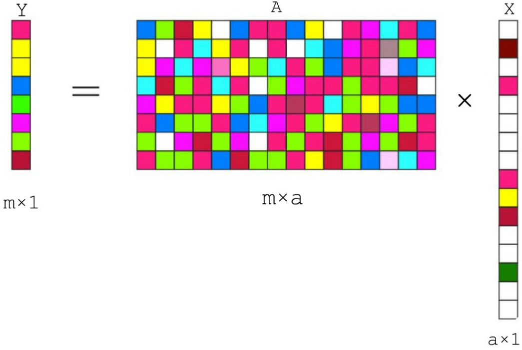
# Dimensionality Reduction (DR)

The projection of high-dimensional data into a low-dimensional space by reducing the input features

*Objectif* : given an instance $X_i \in \mathbb{R}^a$, we wish to obtain $Y_i \in \mathbb{R}^m$, where $m \ll a$

- Principal Component Analysis (PCA)
- **Compressed Sensing (CS)**
- Hashing Trick (HT)
- ...

# Compressed Sensing (CS)




- Data compression method that transforms and reconstructs data from few samples with h.p
- Matrix $A$ used to transform instances from $\mathbb{R}^a \rightarrow \mathbb{R}^m$, $m \ll a$
  - Fourier transform, random matrices (e.g., Bernoulli, Gaussian)

Donoho, Compressed sensing, 2006.

# Compressed Sensing (CS)

CS relies on two principles :

- ***Sparsity*** : expresses the idea that data may be much smaller and are compressible. $X$ is $s$-sparse if $\|X\|_0 \leq s$

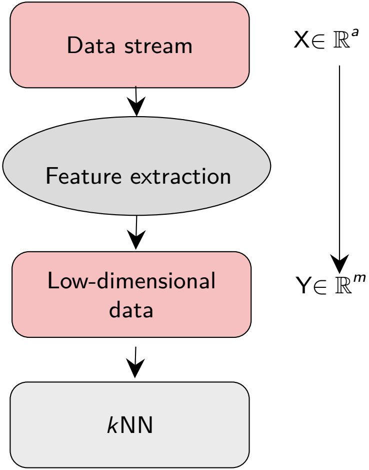
# Compressed Sensing (CS)

CS relies on two principles :

- ***Sparsity*** : expresses the idea that data may be much smaller and are compressible. $X$ is $s$-sparse if $\|X\|_0 \leq s$
- ***Restricted Isometry Property (RIP)*** : $A$ satisfies RIP $\forall$ $s$-sparse instance $X \in \mathbb{R}^a$, if there exists $\epsilon \in [0, 1]$ :

$$(1 - \epsilon)\|X\|_2^2 \leq \|AX\|_2^2 \leq (1 + \epsilon)\|X\|_2^2$$

# Compressed $k$-Nearest Neighbors (CS-$k$NN) [ECAI'20]



# CS-$k$NN : Theoretical Guarantees

The distance between two instances $X_i$ and $X_j$ is defined as follows :

$$D_{X_j}(X_i) = \sqrt{\|X_i - X_j\|^2}$$

The $k$-nearest neighbors distance is defined as :

$$D_{w,k}(X_i) = \min_{\binom{w}{k}, X_j \in w} D_{X_j}(X_i)$$

# CS-$k$NN : Theoretical Guarantees

The distance between two instances $X_i$ and $X_j$ is defined as follows :

$$D_{X_j}(X_i) = \sqrt{\|X_i - X_j\|^2}$$

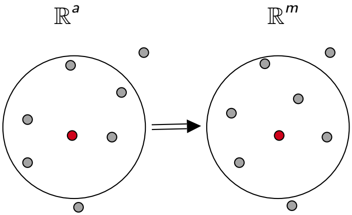
The $k$-nearest neighbors distance is defined as :

$$D_{w,k}(X_i) = \min_{\binom{w}{k}, X_j \in w} D_{X_j}(X_i)$$

**Theorem**

Given a stream $S = \{X_i\}$ and $\epsilon \in [0, 1]$, if there exists a transformation matrix $A : \mathbb{R}^a \to \mathbb{R}^m$ having the RIP, such that $m = \mathcal{O}(s \log(a))$, where $s$ is the sparsity of data, then $\forall X_i \in w$ :

$$(1 - \epsilon) D^2_{w,k}(X) \leq D^2_{w,k}(AX) \leq (1 + \epsilon) D^2_{w,k}(X)$$



# Overview of the data

| Dataset | #Instances | #Attributes | #Classes | Type |
|---|---|---|---|---|
| $\text{Tweets}_1$ | 1,000,000 | 500 | 2 | Synthetic |
| $\text{Tweets}_2$ | 1,000,000 | 1,000 | 2 | Synthetic |
| $\text{Tweets}_3$ | 1,000,000 | 1,500 | 2 | Synthetic |
| RBF | 1,000,000 | 200 | 10 | Synthetic |
| CNAE | 1,080 | 856 | 9 | Real |
| Enron | 1,702 | 1,000 | 2 | Real |
| IMDB | 120,919 | 1,001 | 2 | Real |
| Spam | 9,324 | 39,916 | 2 | Real |
| Covt | 581,012 | 54 | 7 | Real |

# Results

***Accuracy (%)***

| Dataset | CS-$k$NN | HT-$k$NN | PCA-$k$NN | $k$NN |
|---|---|---|---|---|
| $\text{Tweet}_1$ | 78.82 | 73.77 | **80.43** | 79.80 |
| $\text{Tweet}_2$ | 78.13 | 73.02 | **80.06** | 79.20 |
| $\text{Tweet}_3$ | 76.75 | 72.40 | **81.93** | 78.86 |
| RBF | 98.90 | 19.20 | **99.00** | 98.89 |
| CNAE | 70.00 | 65.00 | **75.83** | 73.33 |
| Enron | 96.02 | 95.76 | 94.59 | **96.18** |
| IMDB | 69.86 | 69.65 | 70.57 | **70.94** |
| Spam | 85.39 | 83.82 | **96.00** | 81.17 |
| Covt | 91.36 | 77.18 | 91.55 | **91.67** |
| *Overall* $\varnothing$ | 82.80 | 69.98 | **85.55** | 83.34 |

# Results

***Memory (MB)***

| Dataset | CS-$k$NN | HT-$k$NN | PCA-$k$NN | $k$NN |
|---|---|---|---|---|
| $Tweet_1$ | **2.52** | **2.52** | 3.03 | 34.64 |
| $Tweet_2$ | **2.52** | **2.52** | 5.97 | 70.97 |
| $Tweet_3$ | **2.52** | **2.52** | 8.84 | 103.19 |
| RBF | **2.52** | **2.52** | 8.86 | 13.18 |
| CNAE | **2.52** | **2.52** | 3.09 | 61.37 |
| Enron | **2.52** | **2.52** | 3.51 | 70.60 |
| IMDB | **2.52** | **2.52** | 8.81 | 70.65 |
| Spam | **2.52** | **2.52** | 245.22 | 1476.11 |
| Covt | **2.52** | **2.52** | 3.02 | 3.47 |
| *Overall* ∅ | **2.52** | **2.52** | 32.26 | 211.57 |

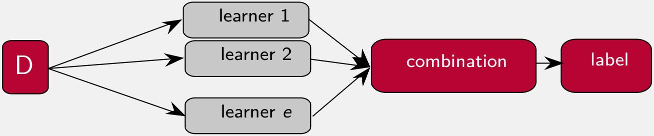
# Results

***Time (sec)***

| Dataset | CS-$k$NN | HT-$k$NN | PCA-$k$NN | $k$NN |
|---|---|---|---|---|
| $\text{Tweet}_1$ | **62.55** | 93.24 | 622.65 | 1198.78 |
| $\text{Tweet}_2$ | **107.48** | 120.83 | 705.71 | 2029.82 |
| $\text{Tweet}_3$ | **126.73** | 154.22 | 988.25 | 2864.55 |
| RBF | **59.47** | 168.31 | 243.26 | 284.34 |
| CNAE | **0.87** | 0.95 | 3.97 | 32.19 |
| Enron | **1.58** | 1.81 | 7.21 | 86.08 |
| IMDB | **95.62** | 125.62 | 1686.88 | 7892.96 |
| Spam | **159.92** | 194.07 | 11329.91 | 34231.45 |
| Covt | **30.94** | 88.17 | 161.00 | 252.69 |
| *Overall* $\varnothing$ | **71.68** | 105.25 | 1749.87 | 5430.32 |

# Ensemble CS-$k$NN (CSB)

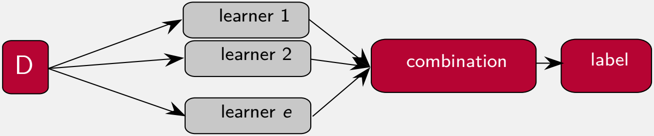
## Ensemble-based method




Bifet, *et al.*, Leveraging bagging for evolving data streams, 2010.

# Ensemble CS-$k$NN (CSB)

**Ensemble-based method**

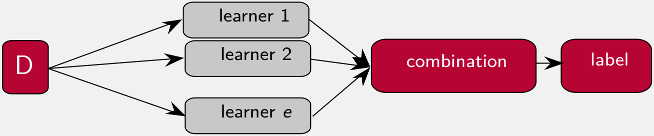



- Uses CS-$k$NN as a base learner under Leveraging Bagging (LB)
- Uses several random matrices : one for each ensemble member
- Preserves the neighborhood properties of the CS-$k$NN

Bifet, *et al.*, Leveraging bagging for evolving data streams, 2010.

# Ensemble CS-$k$NN (CSB)

Ensemble-based method




- Uses CS-$k$NN as a base learner under Leveraging Bagging (LB)
- Uses several random matrices : one for each ensemble member
- Preserves the neighborhood properties of the CS-$k$NN

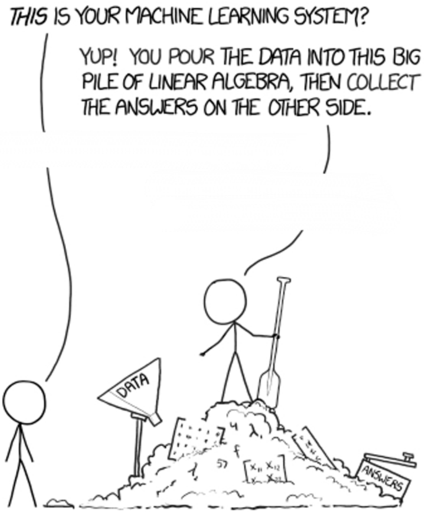
⊕ Good accuracy

⊖ Computational resources

Bifet, *et al.*, Leveraging bagging for evolving data streams, 2010.

# Automated Machine Learning

# Automated Machine Learning

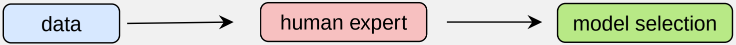


# Automated Machine Learning

# Automated Machine Learning

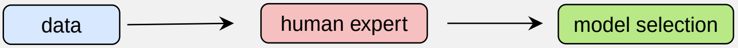
# Automated Machine Learning

## Current practice

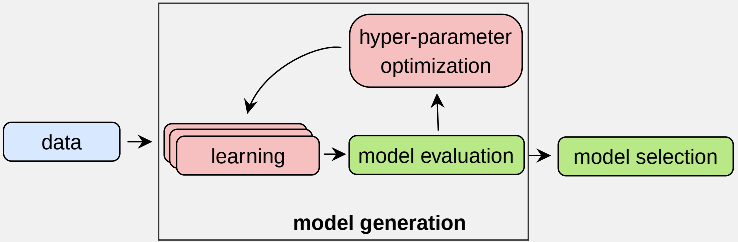


# Automated Machine Learning

## Current practice




## Replace manual model building by automation



## Evolution-Based Online Automated Machine Learning [PAKDD'22]

### AutoML for Stream Classification

- Selecting randomly a population from the configuration space
- Ranking from the best/worst performing configurations
- Generating a new configuration to remove the weakest one

Cedric Kulbach
*(FZI Research)*

Albert Bifet
*(Télécom Paris & University of Waikato)*

Jacob Montiel
*(University of Waikato)*

# Automated Machine Learning For Anomaly Detection

**AutoAD [*Submitted*]**

- An automated framework for unsupervised anomaly detection (batch setting)
- Given different AD algorithms and their hyper-parameter search space, `AutoAD` gives the anomaly scores based on the performance of each approach

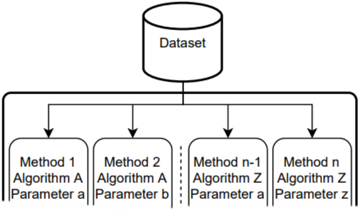
Mauro Sozio

Andrian Putina

Flavia Salutari

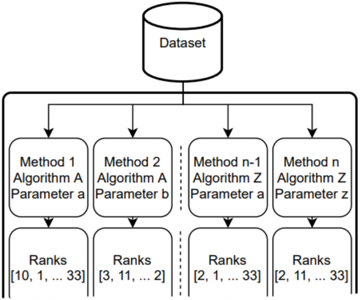
*(Télécom Paris & Huawei France)*

# AutoAD



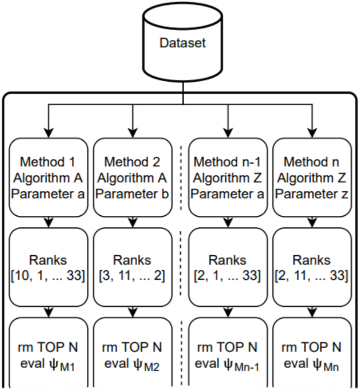

Given a set of methods

# AutoAD



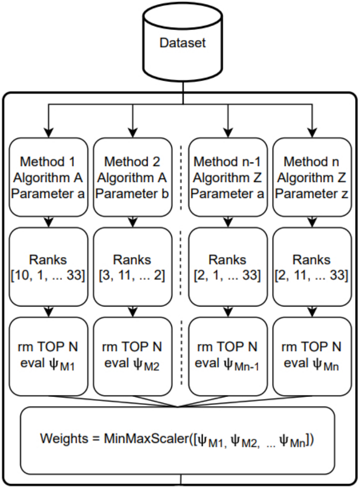

Rank the output anomaly score

# AutoAD



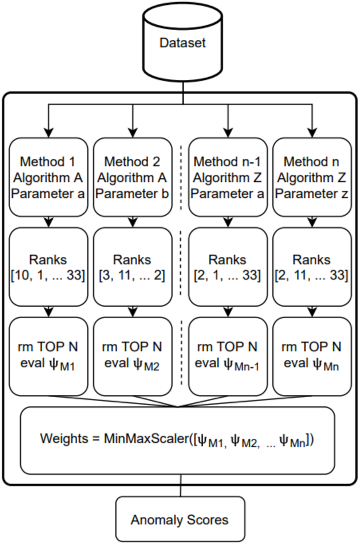

Remove the top anomalous instance
Evaluate the performance of each method

# AutoAD




A weight proportional to the measure is assigned to each method

# AutoAD




Final scores are computed based on the initial scores and the weight assigned to each method

# Thank You !

https://sites.google.com/site/bahrimarouaa/

https://github.com/marouabahri/