Current Trends in Learning from Data Streams

João Gama[1]

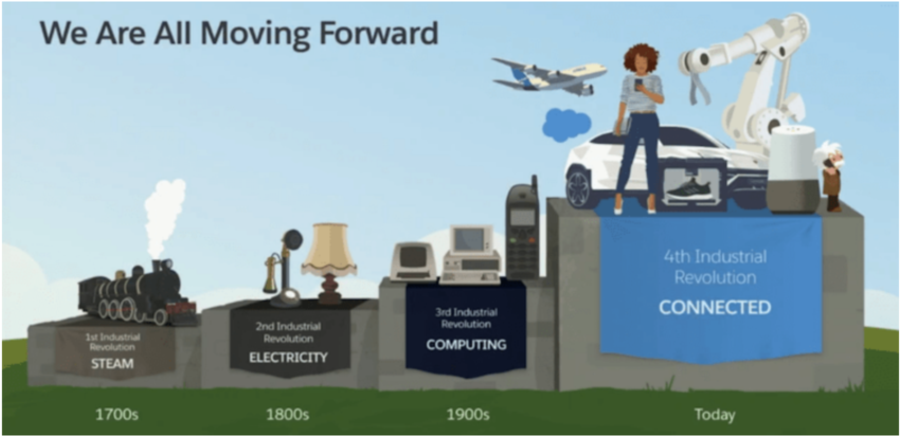
[1]Faculty of Economics, University of Porto and INESC TEC

December 2023

Table of Contents

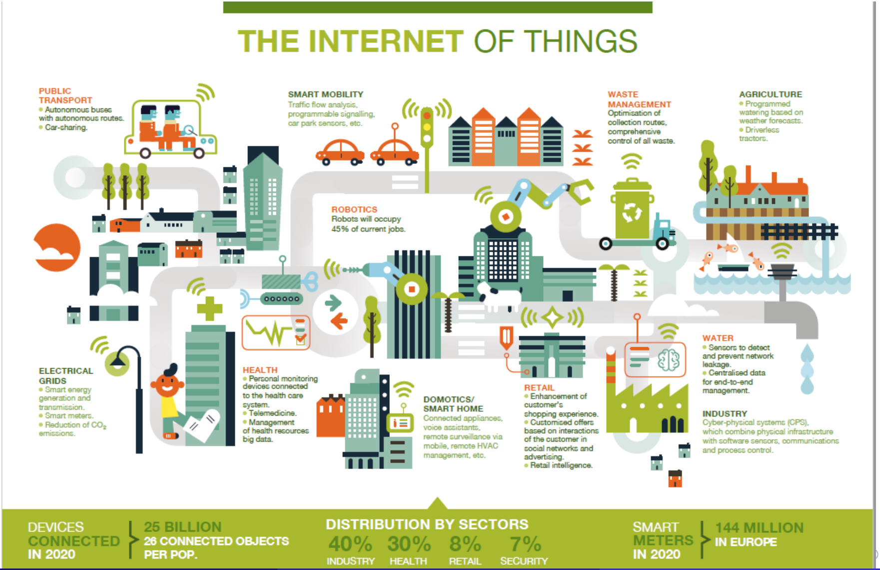
Table of Contents

The 4th Industrial Revolution

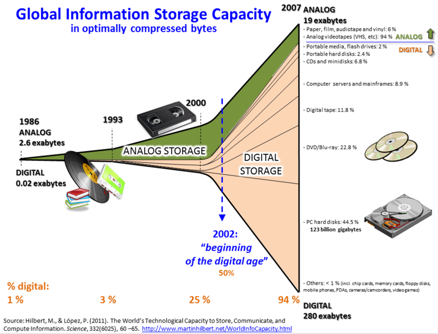


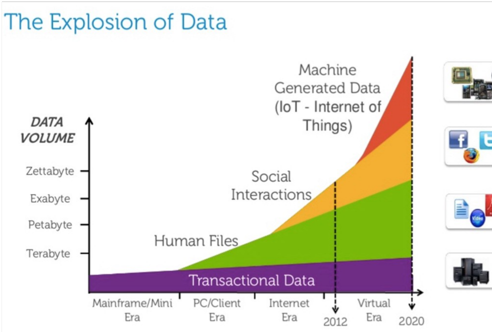
The Internet of Things

We have machines that collect, process, and send information to other machines

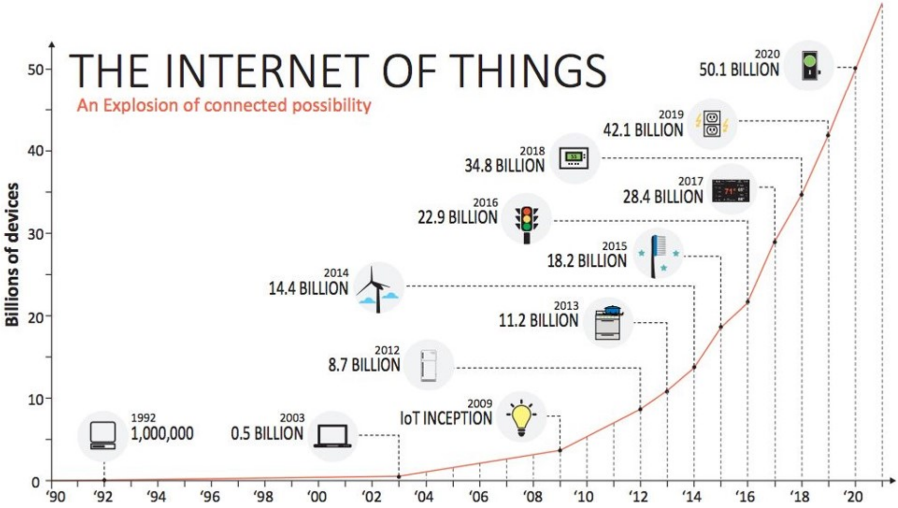


The BigBang of Digital Data



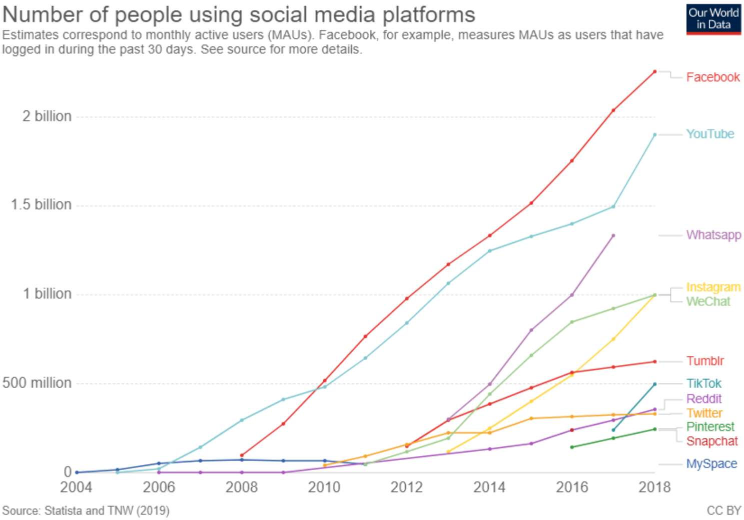


IoT

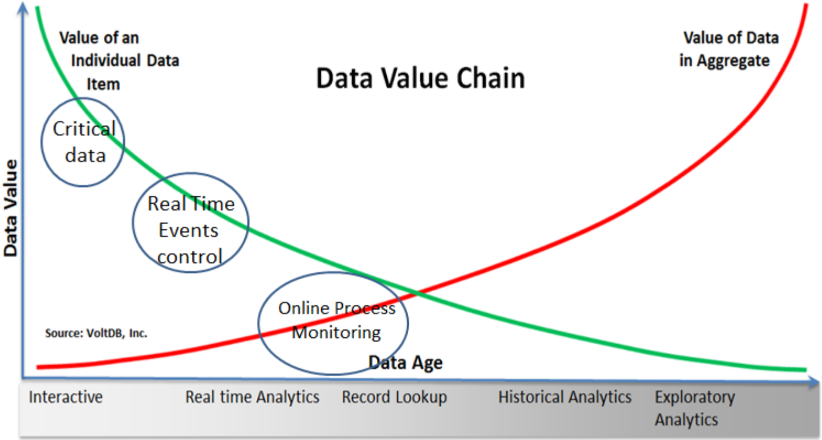


THE INTERNET OF THINGS
An Explosion of connected possibility
Billions of devices
1992
1,000,000
2003
0.5 BILLION
2009
IoT INCEPTION
2012
8.7 BILLION
2013
11.2 BILLION
2014
14.4 BILLION
2015
18.2 BILLION
2016
22.9 BILLION
2017
28.4 BILLION
2018
34.8 BILLION
2019
42.1 BILLION
2020
50.1 BILLION
0
10
20
30
40
50
'90
'92
'94
'96
'98
'00
'02
'04
'06
'08
'10
'12
'14
'16
'18
'20

Social Media



The Value of Data ...



A World in Movement

- The new characteristics of data:
  - **Time and space**: The objects of analysis exist in time and space. Often, they are able to move.
  - **Dynamic environment**: The objects exist in a dynamic and evolving environment.
  - **Information processing capability**: The objects have limited information processing capabilities
  - **Locality**: The objects know only their local spatio-temporal environment;
  - **Distributed Environment**: Objects will be able to exchange information with other objects.
- Main Goal:
  - **Real-Time Analysis**: decision models have to evolve in correspondence with the evolving environment.

Data Streams

Data Streams: Continuous flow of data generated at **high-speed** in **dynamic**, **time-changing** environments.
We need to maintain **decision models** in **real time**.
Learning algorithms must be capable of:

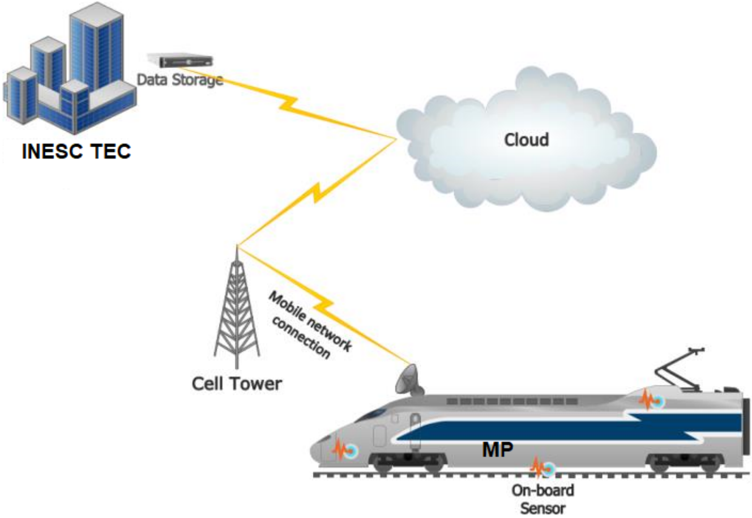
- **incorporating** new information at the speed data arrives;
- **detecting** changes and **adapting** the decision models to the most recent information.
- **forgetting** outdated information;

Unbounded training sets, dynamic models.

Table of Contents

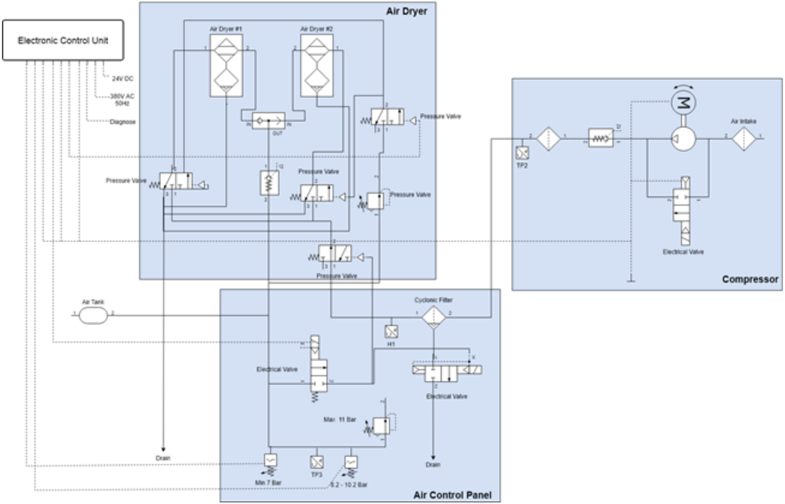
The Context

Real-time failure detection and explanation.



Predictive Maintenance, Adversarial Autoencoders and Explainability M Silva, B Veloso, J Gama, ECMLPKDD 2023

The Air Compressor Unity

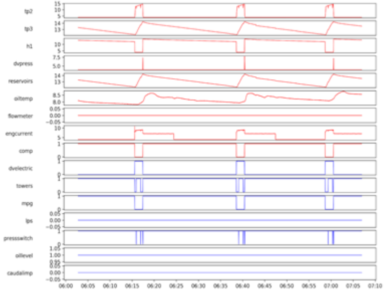


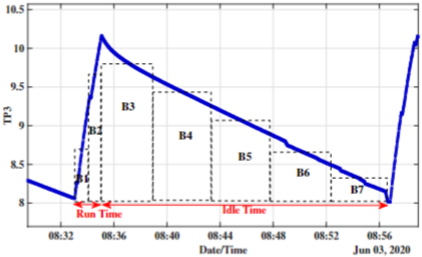
The Air Compressor Unity Sensors

Table 1: Onboard sensors from APU train [19].

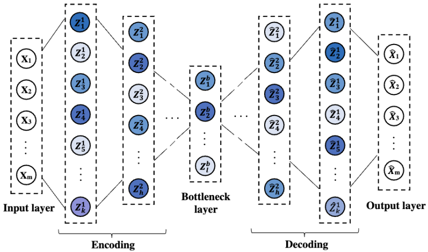
nr.	Module	Description
Analogue		
1	Compressor	TP2 - Compressor Pressure
2	Air Control Panel	TP3 - Pneumatic panel Pressure
3	Air Control Panel	H1 - Pressure above 10.2 Bar
4	Air Dryer	DV - Air Dryer Tower Pressure
5	Air Control Panel	Reservoirs - Pressure
6	Compressor	Oil Temperature
7	Air Control Panel	Flow meter
8	Compressor	Motor Current
Digital		
9	Electronic Control Unit	COMP - Compressor on/off
10	Electronic Control Unit	DV electric - Compressor outlet valve
11	Electronic Control Unit	Towers - Active tower number
12	Electronic Control Unit	MPG - Pressure below 8.2 Bar
13	Electronic Control Unit	LPS - Pressure is lower than 7 bars
14	Electronic Control Unit	Towers Pressure
15	Compressor	Oil Level - Level below min
16	Air Control Panel	Caudal impulses

The Air Compressor Unity Data





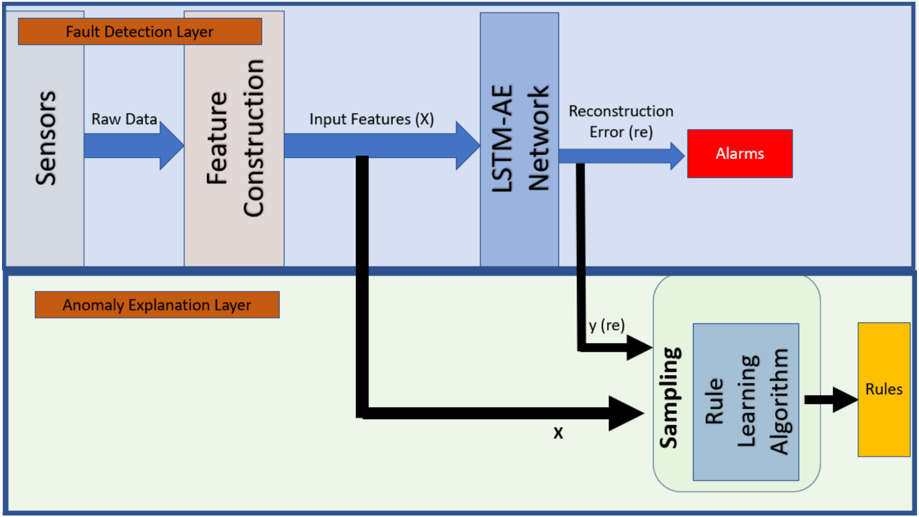
The Fault Detection Layer



- The *Fault Detection layer* is based on a LSTM-AE network trained with normal data. The process is **unsupervised**. [1]
- Each observation is passed through the LSTM-AE and the reconstruction error is computed: $re = \sum_i (x_i - \hat{x}_i)^2$
- High extreme values of the reconstruction error (`re`) is a potential indicator of failures.

[1] S. Maleki, S. Maleki, and N. R. Jennings, "Unsupervised anomaly detection with LSTM-AE using statistical data-filtering," Applied Soft Computing, 2021.

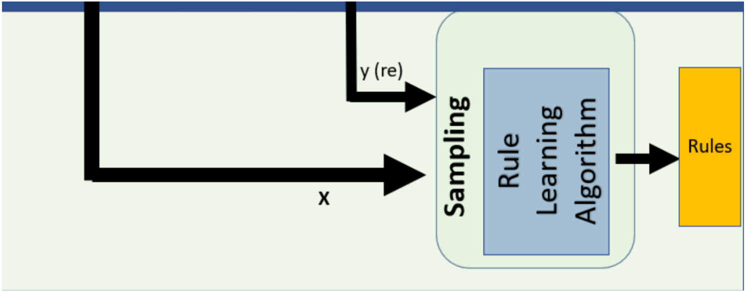
The Neural-Symbolic Explainer

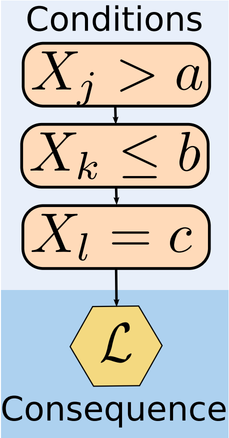


The Anomaly Explanation Layer

The Anomaly Explanation Layer has two main components:

- An online regression rules learning system, based on AMRules. Learns a predictive model $y = f(X)$, where y is the reconstruction error, and X are the input features of the LSTM-AE.
- A sample strategy based on Chebyshev inequality: focusing on the examples with high reconstruction error, meaning high probability of being a failure.





- A rule is an implication of the form *Antecedent* $\Rightarrow$ *Consequent*
- The *Antecedent* is a conjunction of conditions based on attribute values.
- If all the conditions are true, a prediction is made based on *Consequent* ($\mathcal{L}$).
- *Consequent* contains the sufficient statistics to:[a]
    - expand a rule,
    - make predictions,
    - detect changes,

[a] J. Duarte, J. Gama, A. Bifet: Adaptive Model Rules From High-Speed Data Streams. ACM Trans. Knowl. Discov. Data; 2016

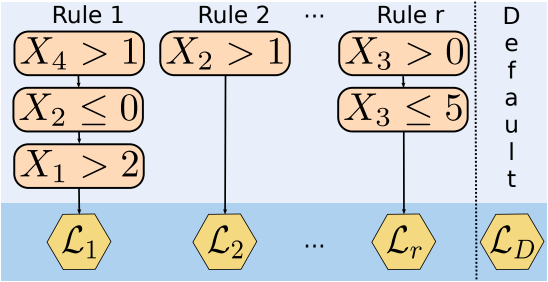
Regression Rules: AMRules

- One-pass algorithm: create, expand, and delete rules online
- Rule expansion: select the literal that most reduce variance of the target
- Uses the Hoeffding bound to decide how many observations are required to create/expand a rule
  - Hoeffding bound $\epsilon = \sqrt{R^2 ln(1/\delta)/(2n)}$
  - Expand when $\sigma_{1st}/\sigma_{2nd} < 1 - \epsilon$
- Evict rule when P-H signals an alarm

```
Input:  S: Stream of examples
begin
  R ← {}, D ← 0
  foreach (X, y) ∈ S do
    foreach Rule r ∈ R do
      if ¬IsAnomaly(X, r)
       then
        if PHTest(error_r,
         λ) then
          Remove the
           rule from R
        end
        else
          Update
           sufficient
           statistics L_r
          ExpandRule(r)
        end
      end
    end
    if S(X) = ∅ then
      Update L_D
      ExpandRule(D)
      if D expanded then
        R ← R ∪ D
        D ← 0
      end
    end
  end
  return (R, L_D)
end
```

Algorithm 1: Training AMRules

Rule sets



- There are two types of rule sets: **unordered** and **ordered**.
- The support $S^u(X)$ of an unordered rule set given X is the set of rules that cover X.
- The support $S^o(X)$ of an ordered rule set is the first rule of $S^u(X)$.
- Given X, only the rules $R_l \in S(X)$ are used for training/testing. The default rule is used if $S(X) = \emptyset$.

Chebyshev inequality

Let Y be a random variable with finite expected value and finite non-zero variance. Then for any real number $t > 0$:

$$P(|y - \bar{y}| \geq t \times \sigma) \leq \frac{1}{t^2}$$

- The probability that an observation is far from the mean is more than $t \times \sigma$ is less than $\frac{1}{t^2}$
- This probability is high for observations near the mean, and low for the observations far away from the mean.
- Those with low probability are the interesting cases: the rare cases -**the failures** [2].

[2] E. Aminian, R. P. Ribeiro, J. Gama: Chebyshev approaches for imbalanced data streams regression models. Data Min. Knowl. Discov. 2021

Chebyshev Over-sampling

For each example:

- the example is presented exactly $K = \left\lceil \frac{|y - \bar{y}|}{\sigma} \right\rceil$.
- K has greater values for the rare cases.

Local Model - 1

Example 1: Air leak failure

```
Sample 4089 re=2941.77 2/21/2021 15:48
Rule 0: B6_H1 > 25663.70
```

This is a failure on the control system of the APU, due to a malfunction of a pneumatic control valve the system opens the escape valves (H1) when the compressor is trying to fill the tanks.

Local Model - 2

Example 2: Oil leak failure

```
Sample 5428 re=1124.203   3/10/2021 21:49
Rule 0: dig7 > 2258.00
```

This is a severe failure due to oil leak. The train driver did not receive any alarm to return to maintenance and the motor seized.

Global Model

- Rules related with oil leak

```
Rule 0: If dig7 > 2258.0 Then 219.2
Rule 1: If dig7 > 2187.0 Then 42.9
```

- Rules air leak located after the pneumatic control panel

```
Rule 2: If B1_TP3 > 7345.6 and B5_MC > 1925.7 Then 1.8
Rule 3: If dig8 > 251.0 Then 2.4
Rule 4: If B6_TP3 < 5635.1 Then 2.5
Rule 5: If B2_H1 > 378.1 Then 1.9
```

Lessons Learned

- The Neural-Symbolic Explainer (NSE) is the first explainer specifically designed for explaining **anomalies**.
- NSE uses two layers:
  - The *Detection layer* is based on state-of-the-art black-box anomaly detection model: LSTM-AE. Unsupervised learning to detect abnormal observations.
  - The *Explanation layer* is based on a transparent model: regression rules. It learns a mapping from the input features to the reconstruction error of the LSTM-AE. Supervised learning to model the LSTM-AE.
  - Both layers run online and in parallel. For each observation, the system produces a classification regarding whether it is faulty and the **why** of the LSTM-AE prediction.

Table of Contents

Motivation

Hyperparameter self-tuning for data streams; Veloso, Gama, et al., Inf. Fusion, 2021

- Hyper-parameter optimization is the problem of choosing a set of optimal hyper-parameters values of a learning algorithm for a specific dataset [3].
- Stream-based algorithms have several parameters that requires a tuning process
- Typically, these algorithms are tuned using a initial training step to adjust the model parameters
- The optimal values of the hyper-parameters evolve over time!

[3] A hyper-parameter is a parameter used to control the learning process.

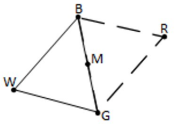
Research Question

- Given
  - A data stream S
  - A learning algorithm A with hyper-parameters $p_1, \ldots, p_n$
  - A loss function L
- Find:
  - the set of hyper-parameter values that minimize the loss function
  - Adapting when concept drift is detected

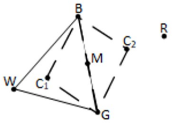
Our approach explores the Nelder & Mead algorithm for function minimisation.

The Nelder & Mead Algorithm

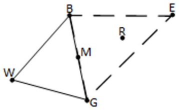
- Optimization algorithm to find a minimum of a function
- Use a simplex with k vertices, where $k = 1 + \mathit{number\ of\ parameters}$ of the function to minimize
- Each vertice corresponds to an instantiation of the hyper-parameters
- Sort the different model configurations by the evaluation metric
- Apply Nelder-Mead Operators to obtain the updated parameters and substitute the Worst Model by the best configuration



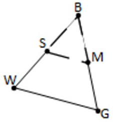
Reflection



Contraction



Expansion



Shrink

Nelder & Mead Algorithm

The vertices are ordered by the evaluation metric:

- best (B),
- good (G), which is the closest to the best vertex,
- worst (W).

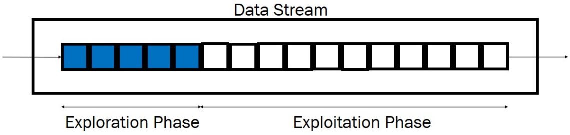
For each Nelder-Mead operation, it is necessary to compute an additional set of vertexes:

- midpoint (M),
- reflection (R),
- expansion (E),
- contraction (C) and
- shrinkage (S)

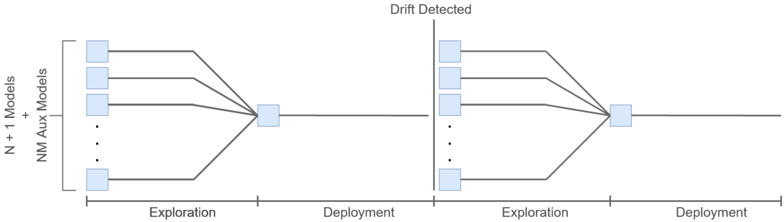
and verify if the calculated vertices belong to the search space.

SPT - Nelder & Mead

- Self Parameter Tuning Algorithm
  - Based on the Nelder-Mead optimisation algorithm
  - Adapted for data streams
  - Is a wrapper over a learning algorithm
- How to estimate the error of a Machine Learning algorithm?
  - Prequential estimation
  - Sample size estimation
- Explore different configurations
  - Parallel computing

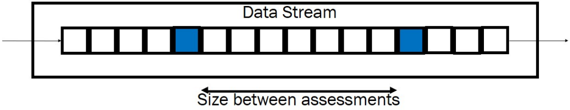


Exploration-Deployment Phases



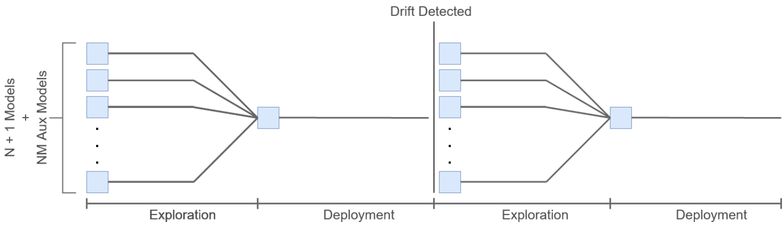
Sample Size Estimation

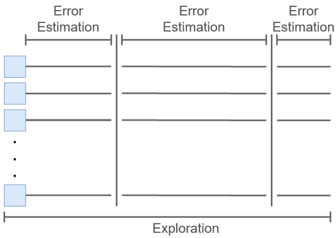
- How many predictions are needed for a fair performance estimation?
- How to select the appropriate moment to apply the Nelder-Mead operators?



- For each model we need to estimate performance for example, estimate the error of a configuration to calculate the sample size:
- $S_{size} = \frac{16\sigma^2}{(1-\delta)^2}$, where S_{size} is the sample size, σ is the standard deviation of the metric and $\delta = 95\%$ is the confidence level

Exploration-Deployment Phases





Experimental Setup

- The SPT approach is compared against the
  - default hyper-parameter initialisation,
  - the grid search algorithm.
- Learning task:
  - Classification
  - Recommendation

We use `prequential` error estimation for measuring performance.

Results - Classification

- Algorithm: EFDT (Extremely Fast Decision Trees)
  C Manapragada, GI Webb, M Salehi
  ACM SIGKDD International Conference on Knowledge, 2018
- Parameters:

	Grace Period	Tie threshold
Default	200	0.05
Grid	[50, 450] incr. 40	[0.01, 0.1] incr. 0.01

- Data set: Electricity, Avila, SEA, Credit
- Evaluation protocol: Prequential
- Evaluation metrics: Error Rate

Results - Classification

Table: Algorithms – Accuracy (%)

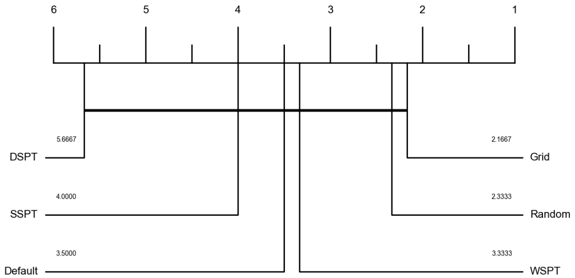
Data set	SPT	Grid Search	Default Parameters
Avila	60.9 (1.00x)	60.8 (0.99x)	56.1 (0.92x)
Credit	80.4 (1.00x)	80.9 (1.01x)	80.0 (0.99x)
Electricity	89.8 (1.00x)	91.9 (1.02x)	82.2 (0.92x)
SEA	88.2 (1.00x)	88.1 (0.99x)	86.6 (0.98x)

Results - Classification

Table: Algorithms – Runtime (ms)

Data set	SPT	Grid Search	Default Parameters
Avila	5636.07 (1.00x)	38 378.40 (6.80x)	389.07 (0.07x)
Credit	10 991.7 (1.00x)	72 698.10 (6.61x)	585.10 (0.05x)
Electricity	14 931.67 (1.00x)	52 702.60 (3.53x)	491.00 (0.03x)
SEA	7377.90 (1.00x)	25 806.57 (3.50x)	314.43 (0.04x)

Critical Difference Diagram: Classification



Hyperparameter Tuning for Recommendation Systems

Hyper-parameter Optimization for Latent Spaces; Veloso & Gama, et al, ECML/PKDD 2021

- Problem: Recommending items to user using matrix faxtorization
- use streaming data to train and validate model using prequential protocol
- Initial Setup: simple embedding model

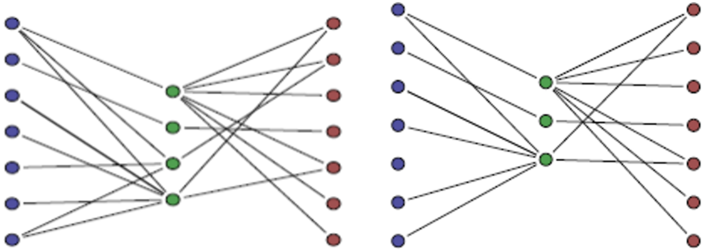
	M1	M2	M3	M4	M5
F1	3	1	1	3	1
F2	1	2	4	1	3

	F1	F2
A	1	0
B	0	1
C	1	0
D	1	1

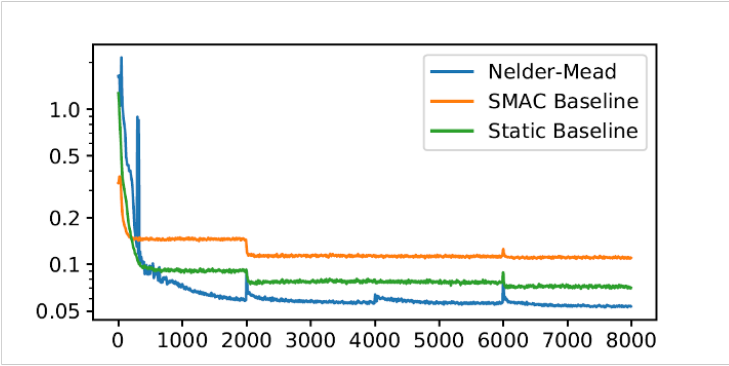
	M1	M2	M3	M4	M5
A	3		1		1
B	1		4	1	
C	3	1		3	1
D		3		4	4

Hyperparameter Tuning for Recommendation Systems

User/Features/Items Graph



Experimental Results



Lessons Learned

- Sound method for hyper-parameter tuning of stream-based classifiers
- Fast convergence
- Outperform the baseline methods

Table of Contents

Current Trends and Open Issues

- Network Data
- Deep models for data streams
- Evolving Feature Spaces: sensor networks
- Structured Output Prediction: predicting vectors, trees, graphs, ...
- Open World Machine Learning: novelty detection, open set recognition
- ...

Conclusions

Learning from Data Streams: An existential pleasure!

Thank you!

Thanks to my collaborators:

- Bruno Veloso
- Rita P. Ribeiro
- Saulo Mastelini
- Shazia Tabassum
- Narges Davari

and Projects FailStopper (FCT), Explaining Predictive Maintenance (CHIST-ERA)