Analyse des signaux temporels (une introduction) avec une application sur les signaux EEG

(Anxious States based on a psychological stimulation)

Ahmed Rebai (02/07/2022)

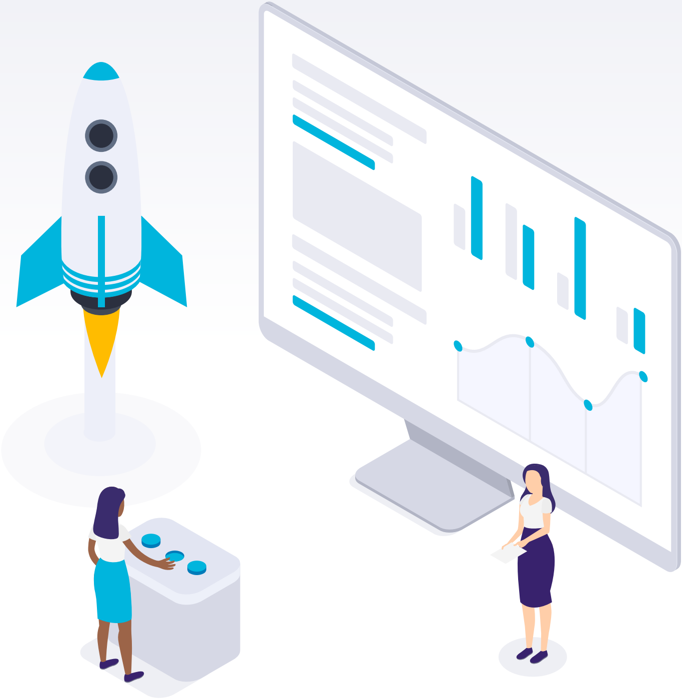
hackerrank.com/ahmed_rebai2
fr.slideshare.net/ahmed_rebai
github.com/AhmedRebai
ahmedrebai.wixsite.com/scientificpython
https://tinyurl.com/mr3ru8xf
linkedin.com/in/ahmed-rebai-phd/

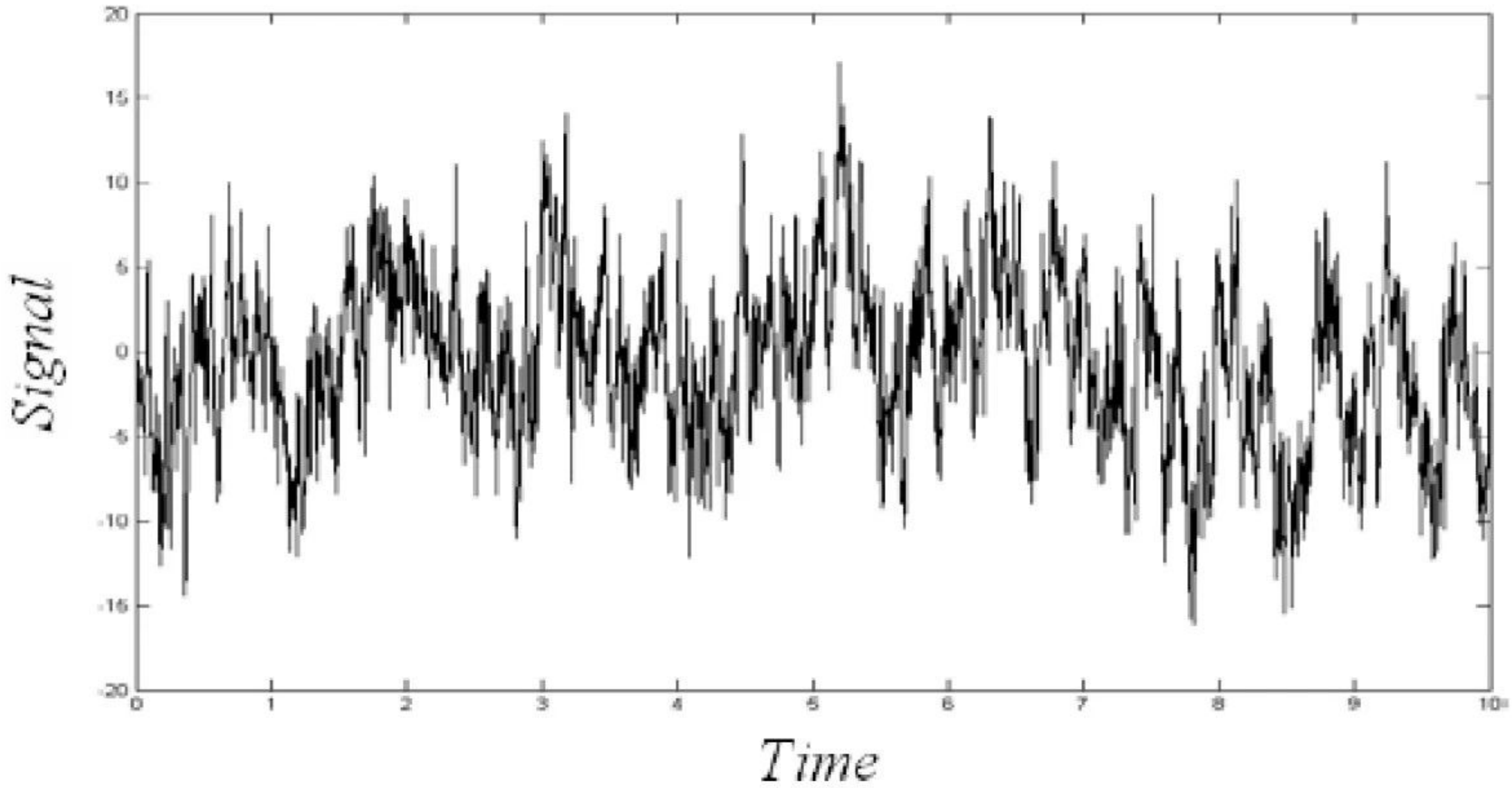
Plan de la présentation

1. Introduction
2. ML vs DL
3. Exemple 1: Prédiction du nombre des visiteurs
4. Compétition: classification des signaux EEG et détection des états de stress
5. Décomposition
6. Ressources pour la compétition
7. Conclusion

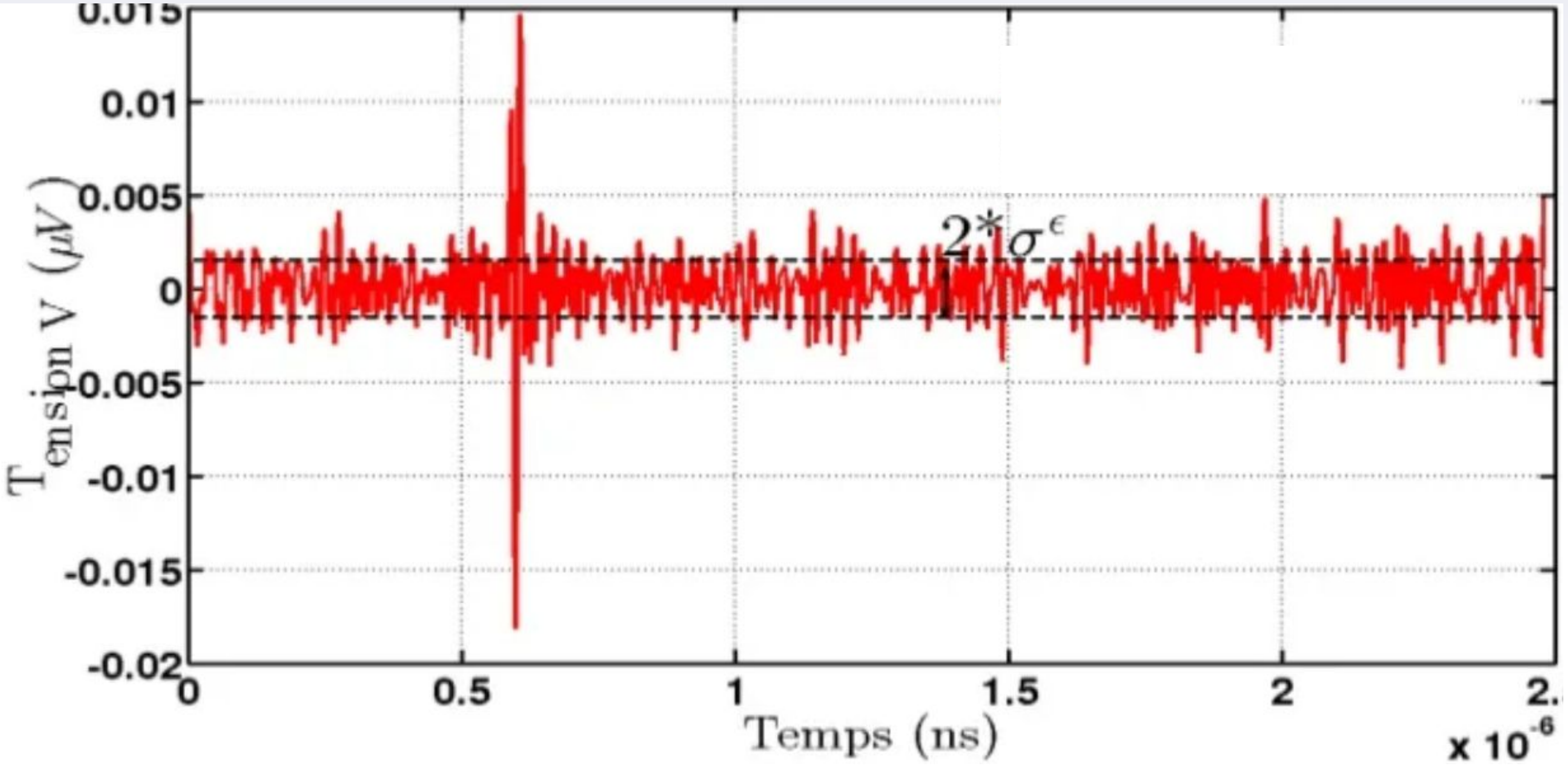
1

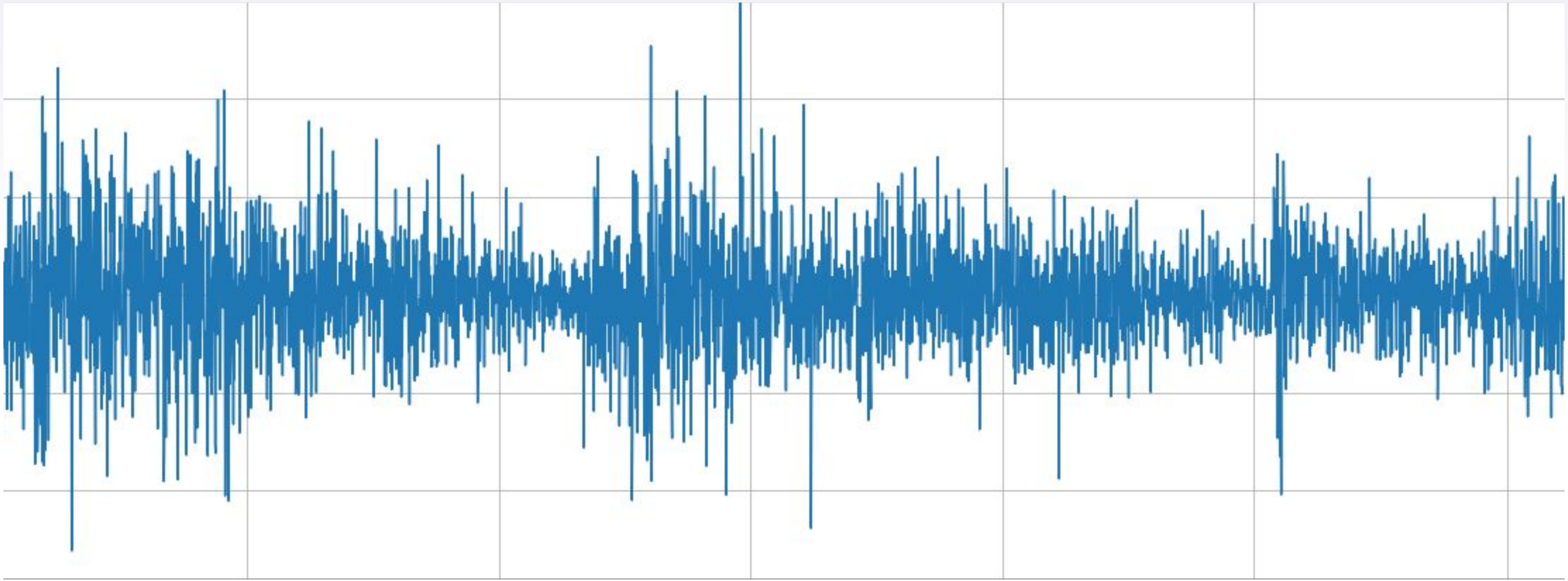
Introduction



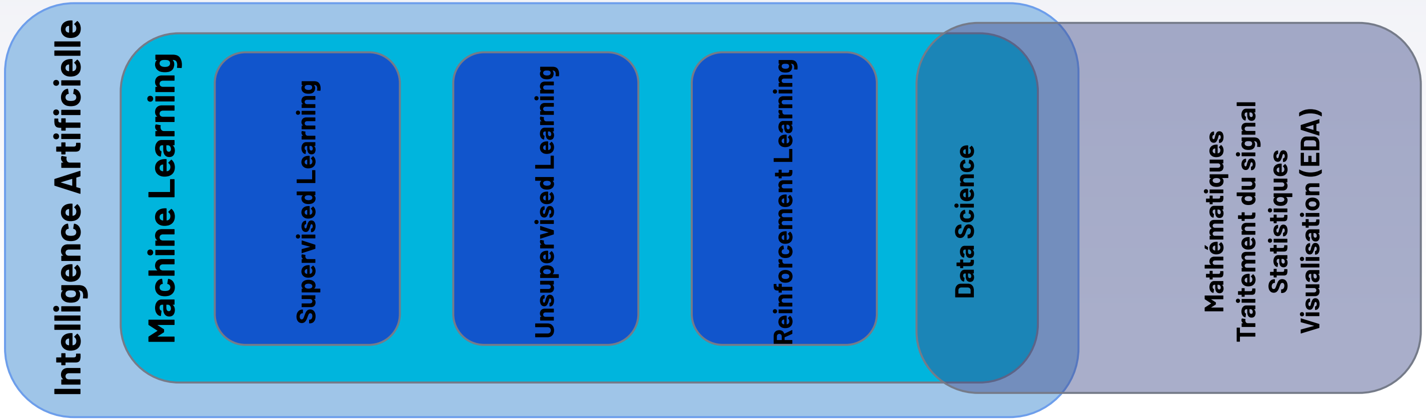


Signal
Time
20
15
10
5
0
-5
-10
-15
-20
0
1
2
3
4
5
6
7
8
9
10


0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
0
0.5
1
1.5
2
Tension V (μV)
Temps (ns)
x 10^-6
$2^{*}\sigma^{\epsilon}$

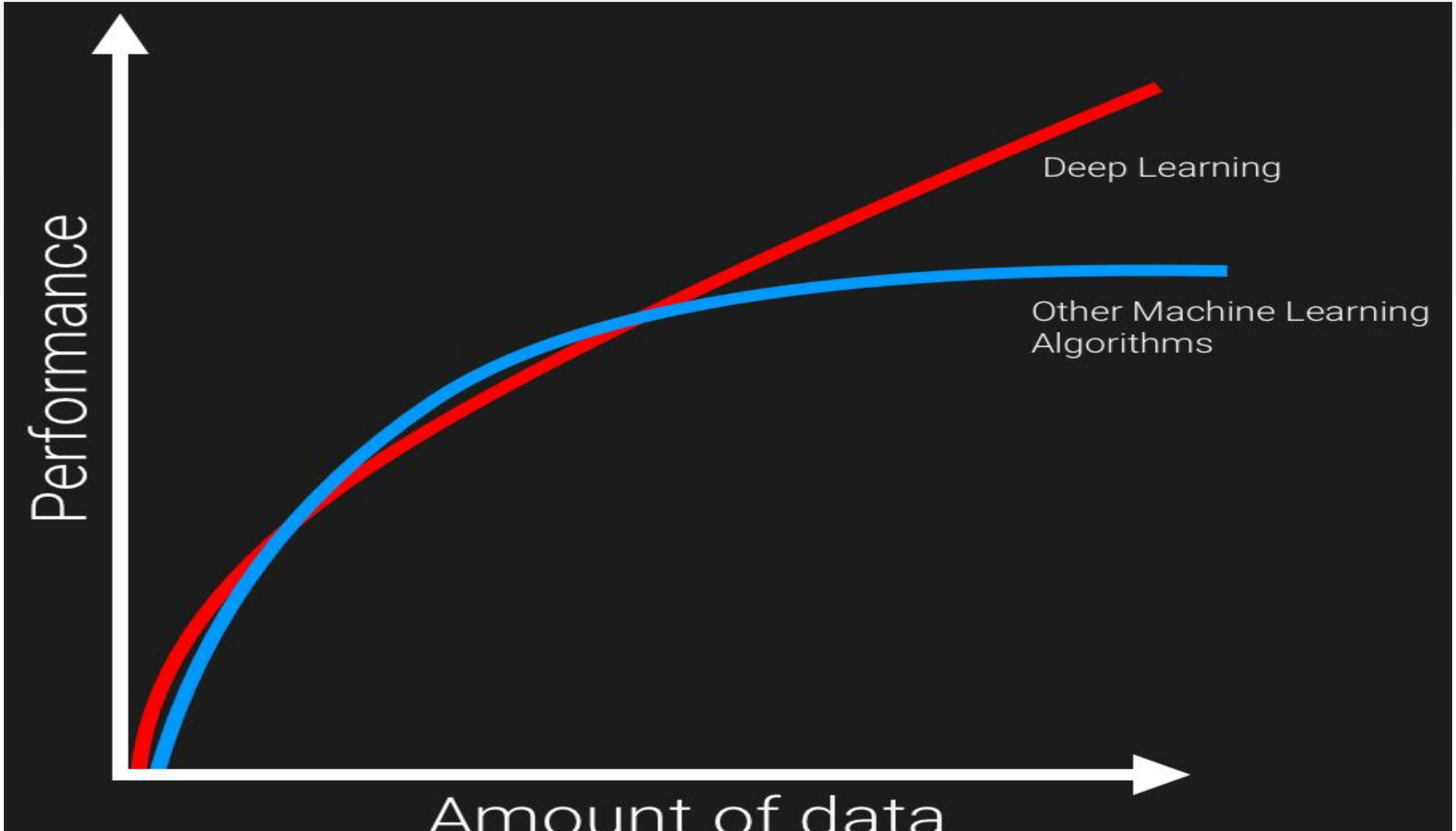


AI, ML, DL, RL, DS… Quelques définitions

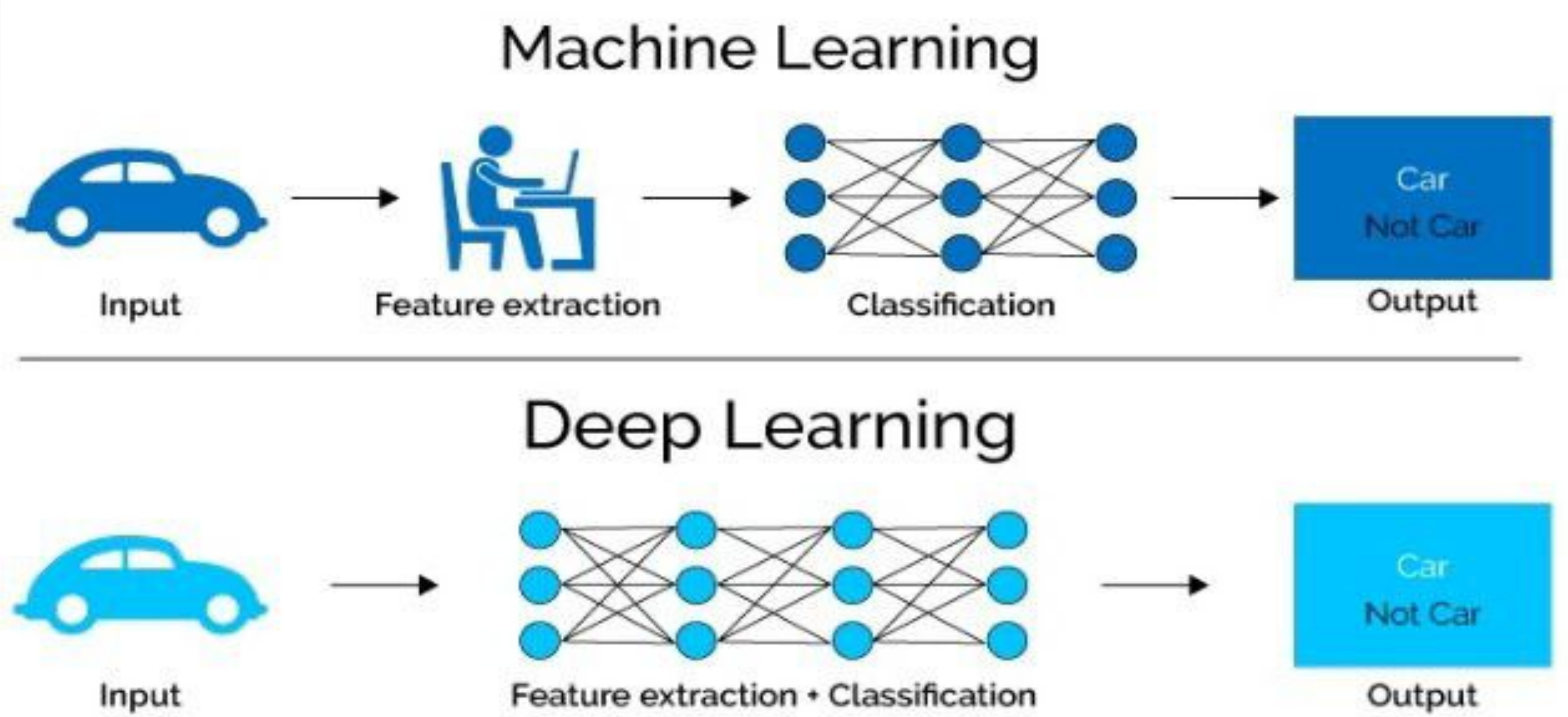


- **L'intelligence artificielle** – permet aux machines d'effectuer des tâches sans programmation préalable.
- **Le machine learning** – est basé sur l'apprentissage statistique paramétrique ou non-paramétrique à partir des données ainsi il permet de réaliser des prédictions.
- **Le deep learning** – est une branche du machine learning utilisant des réseaux de neurones.
- **Le reinforcement learning** – est la seule branche où l'IA pourrait dépasser celle de l'humain.

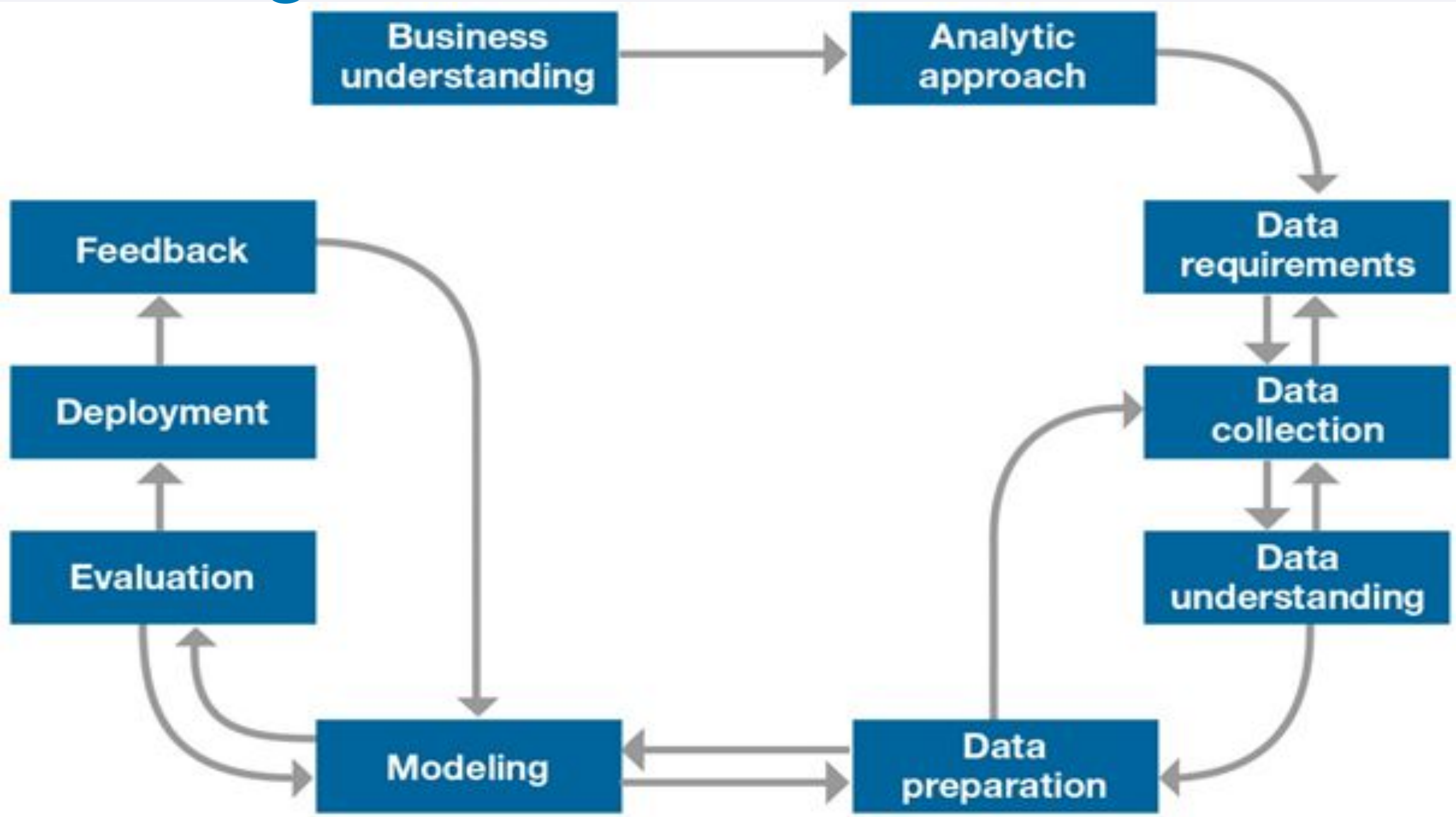
ML vs DL


Deep Learning
Other Machine Learning Algorithms
Performance
Amount of data

ML vs DL


Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output
Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output

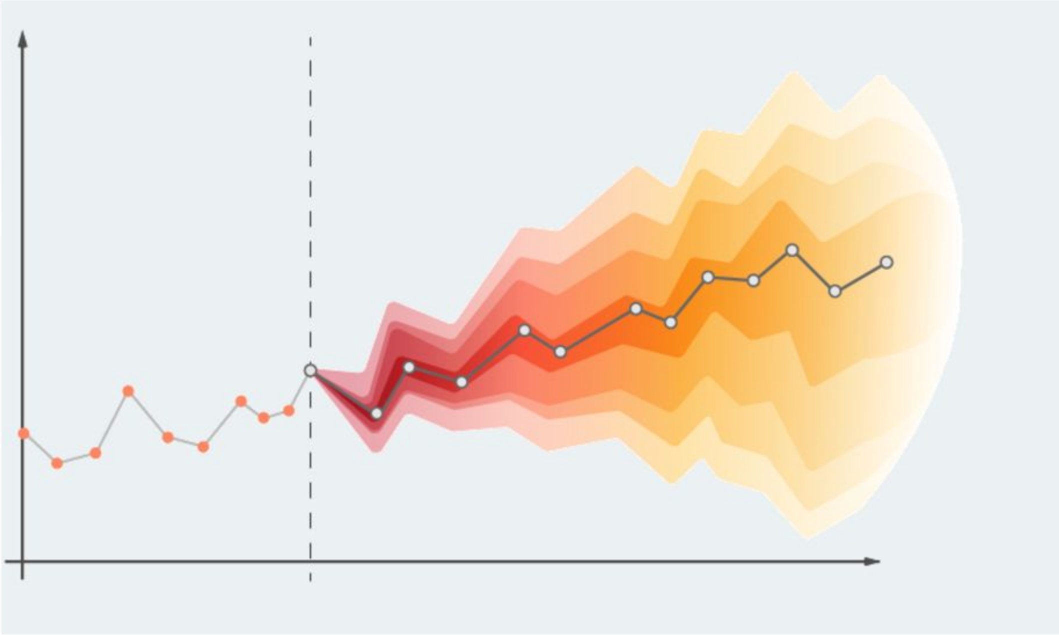
Méthodologie



2

Exemple 1:
Prédiction du nombre des visiteurs

Prédiction du nombre des visiteurs



Compréhension des données

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
202930	1	5	2015-01-30	5577	616	1	1	0	0
202931	2	5	2015-01-30	5919	624	1	1	0	0
202932	3	5	2015-01-30	6911	678	1	1	0	0
202933	4	5	2015-01-30	13307	1632	1	1	0	0
202934	5	5	2015-01-30	5640	617	1	1	0	0
...	...	...	...	...	...	...	...	...	...
204040	1111	5	2015-01-30	5351	453	1	1	0	0
204041	1112	5	2015-01-30	10099	784	1	1	0	0
204042	1113	5	2015-01-30	7905	778	1	1	0	0
204043	1114	5	2015-01-30	24489	3338	1	1	0	0
204044	1115	5	2015-01-30	9525	580	1	1	0	0

Description des variables

- Le tableau contient les données de plusieurs magasins dont nous avons choisi un au hasard .
- **DayOfWeek** Indique le jour de la semaine (0 : Sunday , 1 : Monday , 2 : Tuesday etc...)
- **Date** indique la date (year - month - day format)
- **Sales** : le chiffre d'affaires pour un jour donné
- **Customers** : le nombre de clients un jour donné
- **Open** :un indicateur permettant de savoir si le magasin était ouvert: 0 = closed, 1 = open
- **StateHoliday** - Le tableau contient les données de plusieurs magasins dont nous avons choisi un au hasard . Normalement, tous les magasins, à quelques exceptions près, sont fermés les jours fériés. Notez que toutes les écoles sont fermées les jours fériés et les week-ends, a = public holiday, b = Easter holiday, c = Christmas, 0 = None
- **SchoolHoliday** - indique si le (magasin, date) a été touché par la fermeture des écoles publiques
- Promo - indique si un magasin organise une promotion ce jour-là

Préparation des données (1/3)

Date	2013-01-08	2013-01-09	2013-01-10	2013-01-11	2013-01-12	2013-01-13	2013-01-14	2013-01-15	2013-01-16	2013-01-17
expanding_median	555	564.5	574	605.5	574	564.5	555	547	539	547
expanding_mean	409.429	443.875	481.778	497.3	501.091	487.417	449.923	455.643	459.467	467.688
rolling_median	555	574	574	637	637	637	637	539	530	530
rolling_mean	409.429	507.286	526.571	538.286	533.286	535.143	535.143	501.857	477.286	449.571
t-7	0	650	555	574	324	0	763	685	785	637
t-6	650	555	574	324	0	763	685	785	637	539
t-5	555	574	324	0	763	685	785	637	539	337
t-4	574	324	0	763	685	785	637	539	337	0
t-3	324	0	763	685	785	637	539	337	0	530
t-2	0	763	685	785	637	539	337	0	530	513
t-1	763	685	785	637	539	337	0	530	513	591

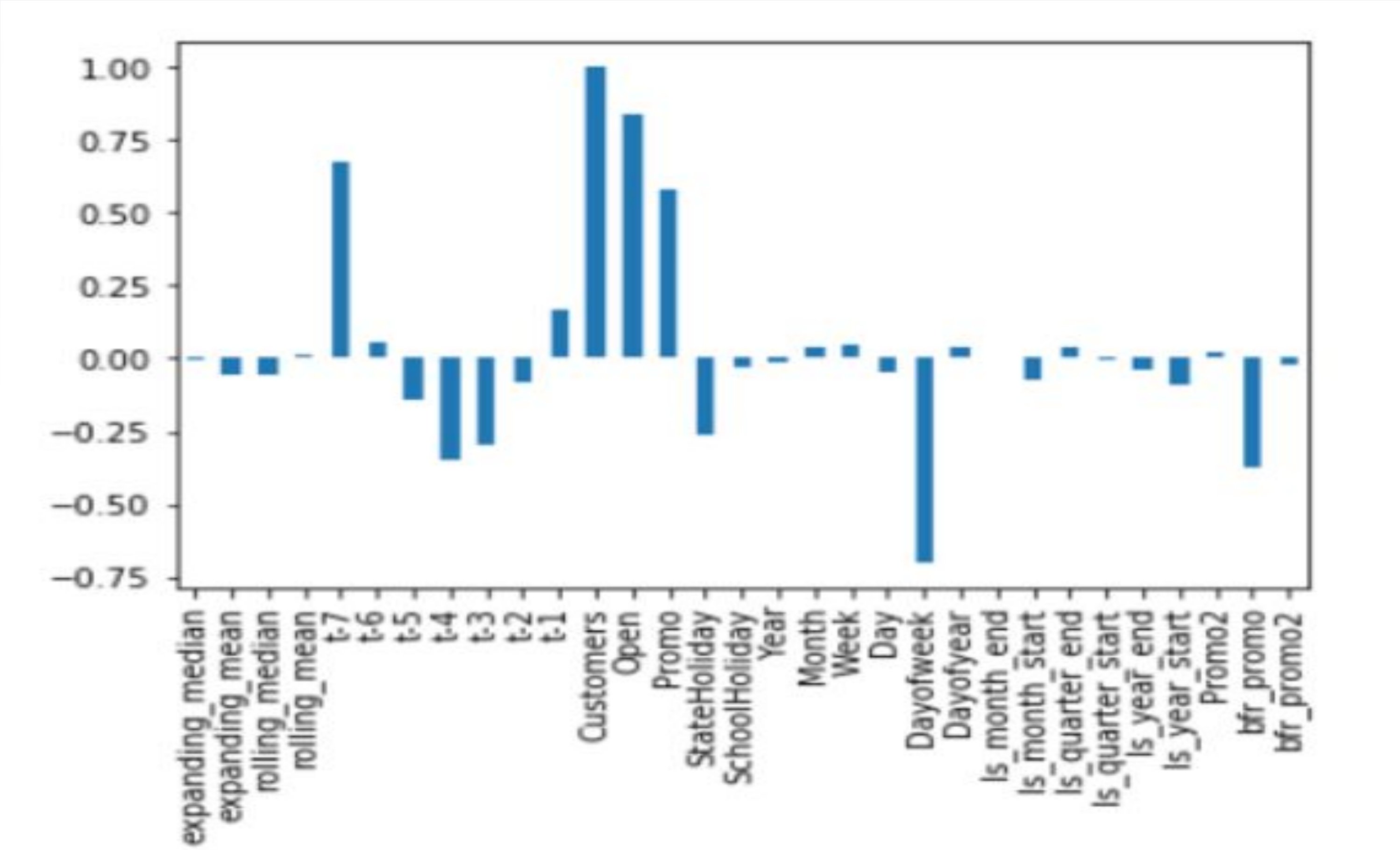
Préparation des données (2/3)

Customers	685	785	637	539	337	0	530	513
Open	1	1	1	1	1	0	1	1
Promo	1	1	1	1	0	0	0	0
StateHoliday	0	0	0	0	0	0	0	0
SchoolHoliday	41	40	39	38	37	36	35	34
Year	2013	2013	2013	2013	2013	2013	2013	2013
Month	1	1	1	1	1	1	1	1
Week	2	2	2	2	2	2	3	3
Day	8	9	10	11	12	13	14	15
Dayofweek	1	2	3	4	5	6	0	1
Dayofyear	8	9	10	11	12	13	14	15
Is_month_end	0	0	0	0	0	0	0	0
Is_month_start	0	0	0	0	0	0	0	0
Is_quarter_end	0	0	0	0	0	0	0	0
Is_quarter_start	0	0	0	0	0	0	0	0

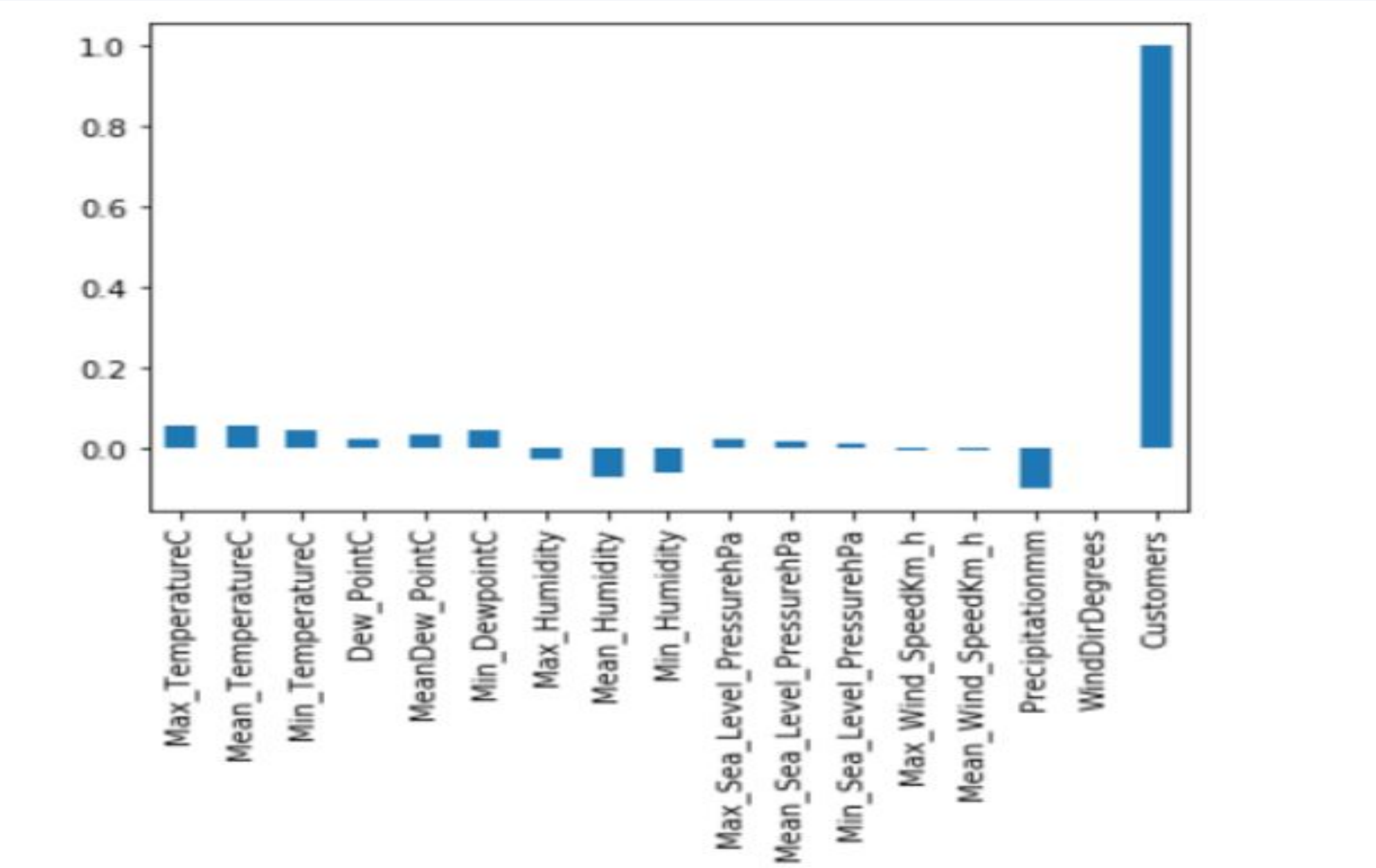
Préparation des données (3/3)

bfr_promo	0	0	0	0	9	8	7	6	5	4	...	5	4
bfr_promo2	0	0	0	0	0	0	0	0	0	0	...	0	0
Max_TemperatureC	5	6	3	-1	-1	-3	-4	-6	-4	-2	...	30	22
Mean_TemperatureC	4	4	1	-2	-4	-6	-6	-9	-6	-4	...	23	19
Min_TemperatureC	4	3	-1	-4	-7	-9	-7	-11	-8	-6	...	17	17
Dew_PointC	5	5	2	-2	-5	-4	-5	-7	-4	-2	...	19	18
MeanDew_PointC	4	4	1	-5	-6	-7	-6	-9	-5	-4	...	17	12
Min_DewpointC	3	2	-2	-7	-8	-10	-8	-13	-7	-7	...	15	10
Max_Humidity	100	100	100	93	93	93	100	100	100	100	...	89	90
Mean_Humidity	92	93	88	79	78	88	92	89	97	96	...	63	59
Min_Humidity	81	81	70	61	51	74	81	76	90	85	...	37	38
Max_Sea_Level_PressurehPa	1027	1024	1015	1019	1021	1021	1019	1013	1013	1018	...	1016	1017
Mean_Sea_Level_PressurehPa	1024	1019	1010	1014	1018	1017	1013	1008	1008	1013	...	1014	1016
Min_Sea_Level_PressurehPa	1022	1013	1006	1009	1016	1015	1010	1006	1006	1010	...	1010	1013
Max_Wind_SpeedKm_h	26	29	35	21	16	21	11	13	16	24	...	26	18
Mean_Wind_SpeedKm_h	13	19	23	14	10	8	6	8	11	11	...	10	11
Precipitationmm	2.03	0	0.51	0	0	0	0.25	0	2.03	3.05	...	0	0

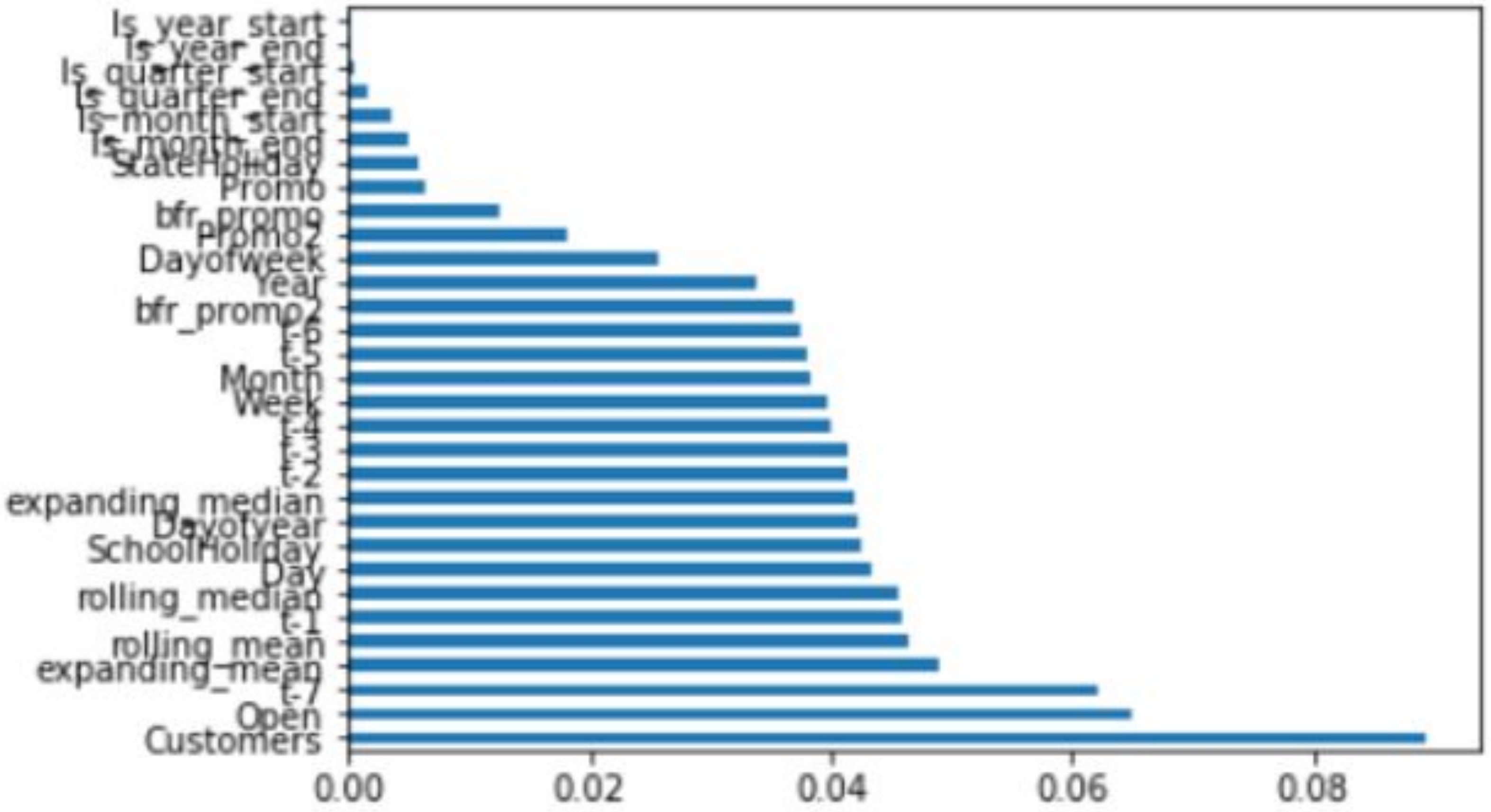
Feature selection: Random Forest feature importance (1/2)



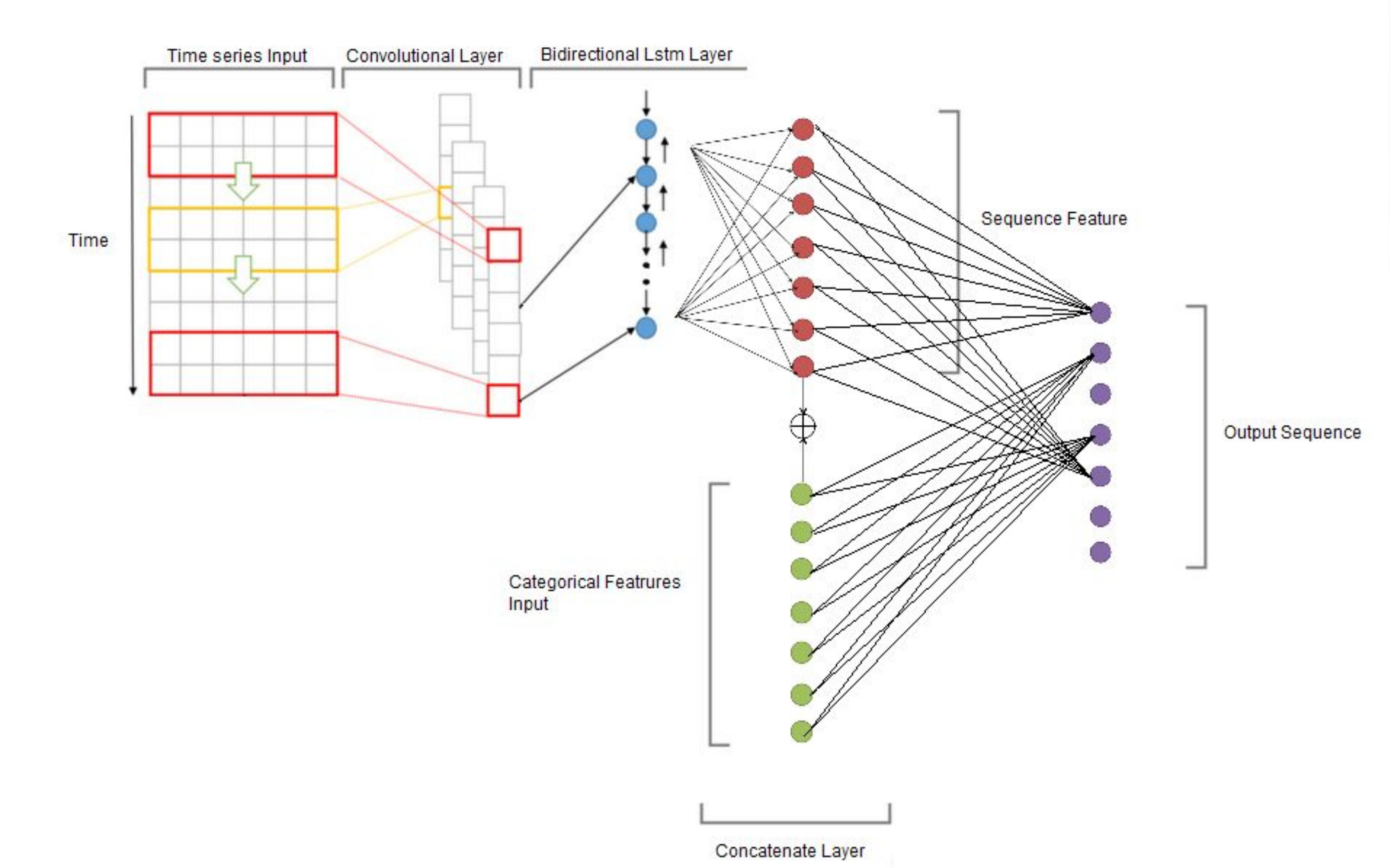
Feature selection: Random Forest feature importance (2/2)



Feature selection: tests de corrélation

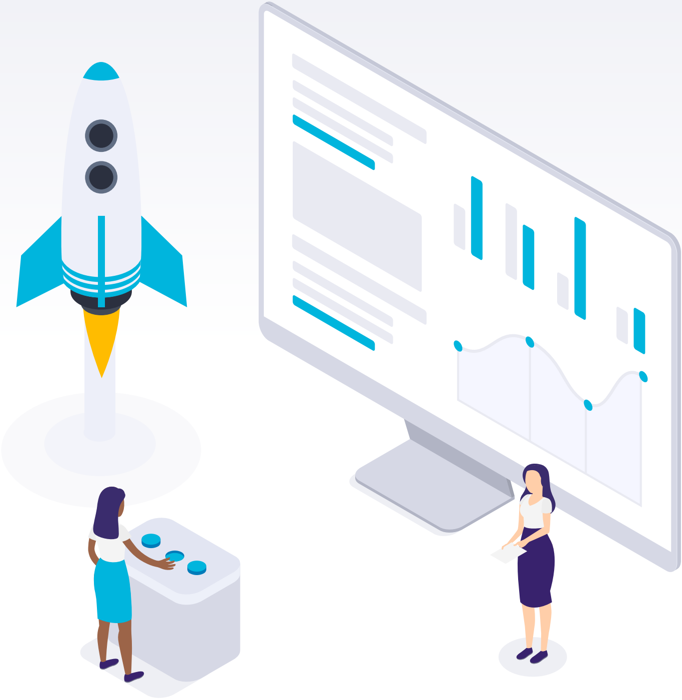


Modélisation

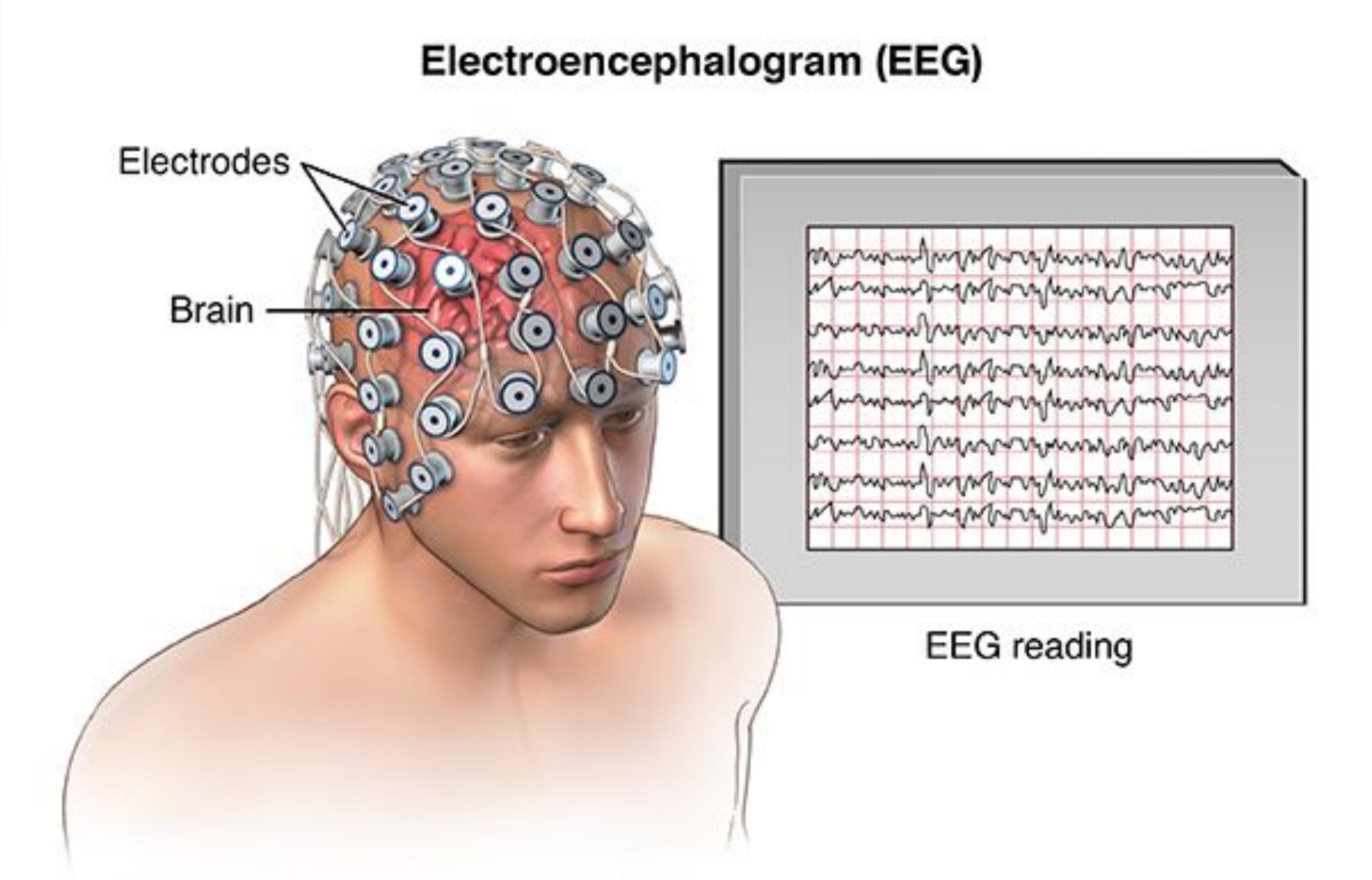


3

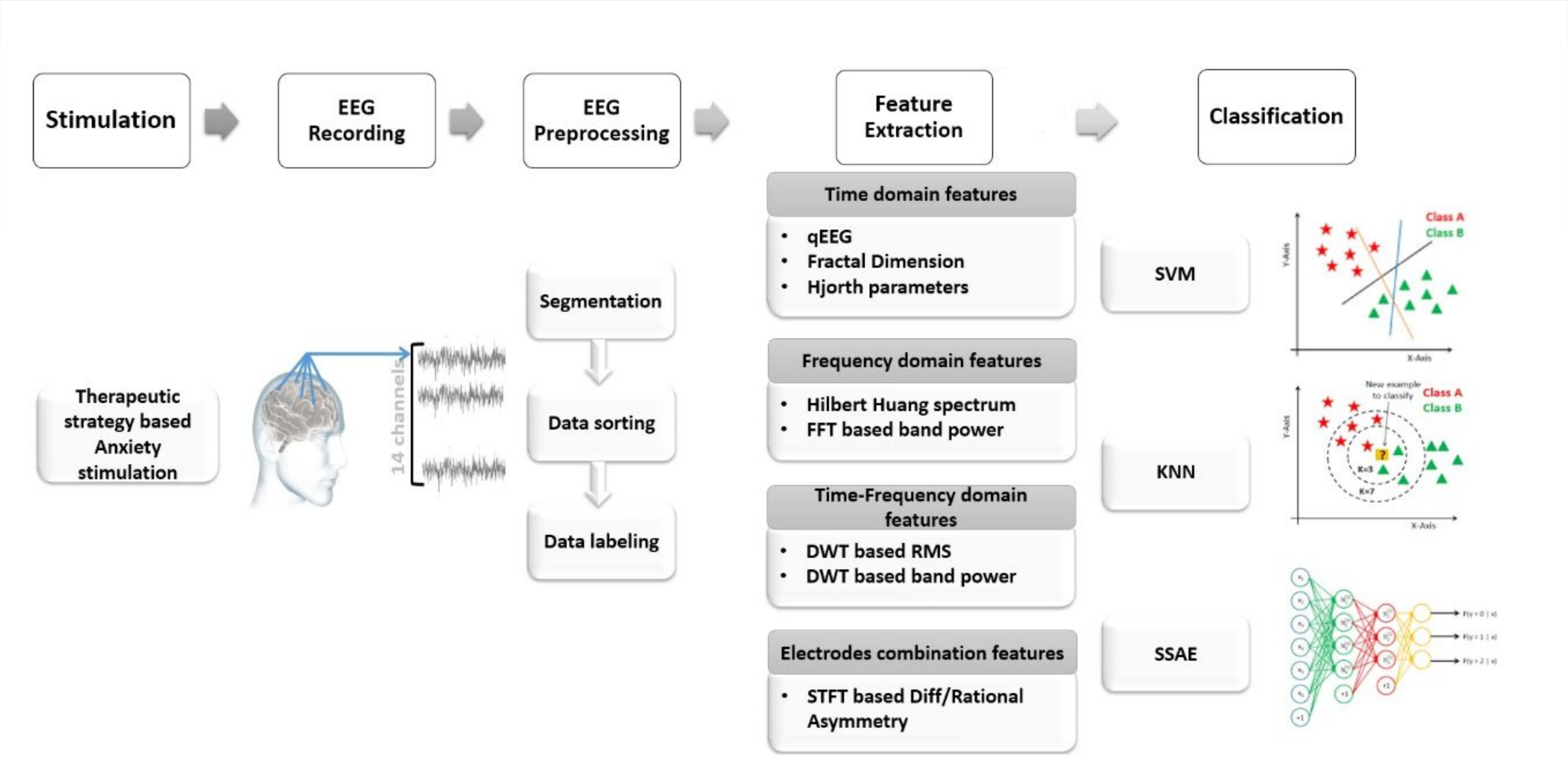
Compétition: Classification des signaux EEG et détection des états de stress



Electroencéphalogramme



Architecture proposée: article scientifique arxiv.org/abs/1901.02942v2



Features extraction dans le domaine temporel

1. **Mean absolute value:** it gives information about muscle contraction levels. It is defined as:

$$MAV_k = \frac{1}{N}\sum_{i=1}^{N}|x_i|$$

2. **Root mean square:** it reflects the mean power of the signal and is related to the constant force and non-fatiguing contractions. It is defined as:

$$RMS_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}{x_i}^2}$$

3. **Variance:** it is a measure of the power density of the signal. It is defined as:

$$VAR_k = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2$$

Features extraction dans le domaine temporel

4. **Simple square integral:** it gives a measure of the energy of the EMG signal. It is defined as:

$$SSI_k = \sum_{i=1}^{N}(\left|x_i{}^2\right|)$$

5. **Zero crossings**: it is the number of times the waveform crosses zero. This feature provides an approximate estimation of frequency domain properties. The threshold avoids counting zero crossings induced by noise. It is calculated as follows

Increment one unit the zero crossings count value if

$$\{x_i > 0 \; and \; x_{i+1} < 0\} \; or \; \{x_i < 0 \; and \; x_{i+1} > 0\} \; and \; \left|x_i - x_{i+1}\right| \geq threshold$$

Which can be rewritten as:

$$ZC_k = \sum_{i=1}^{N-1}\{f(x_i * x_{i+1})\};$$

$$f(x) \;=\; 1 \; if \; x \;<\; 0 \; and \; \left|x_i - x_{i+1}\right| \geq threshold$$

$$f(x) \;=\; 0 \; otherwise$$

Features extraction dans le domaine temporel

6. **Waveform length:** it is the cumulative length of the waveform over the segment. This feature is related to the signal amplitude, frequency and time. It is defined as:

$$WL_k = \sum_{i=1}^{N-1} |x_{i+1} - x_i|$$

7. **Slope sign changes:** it is similar to the zero crossings feature. It also provides information about the frequency content of the signal. It is calculated as follows Increment one unit the slope sign change value if:

$$\{x_i > x_{i-1} \text{ and } x_i > x_{i+1}\} \text{ or } \{x_i < x_{i-1} \text{ and } x_i < x_{i+1}\} \text{ and } |x_i - x_{i+1}| \geq \text{threshold or } |x_i - x_{i-1}| \geq \text{threshold}$$

Which can be rewritten as:

$$SSC_k = \sum_{i=2}^{N-1} [f[(x_i - x_{i-1})(x_i - x_{i+1})]];$$

$$f(x) = 1 \text{ if } x \geq \text{threshold}$$

$$f(x) = 0 \text{ otherwise}$$

8. **Willison amplitude:** it is the number of times that the difference in amplitude between adjacent data points exceed a predefined threshold. This feature provides information about the muscle contraction level. It is defined as:

$$WAMP_k = \sum_{i=1}^{N-1} f(|x_i - x_{i+1}|)$$

$$f(x) = 1 \text{ if } x \geq \text{threshold}$$

$$f(x) = 0 \text{ otherwise}$$

Features extraction

6. **Waveform length:** it is the cumulative length of the waveform over the segment. This feature is related to the signal amplitude, frequency and time. It is defined as:

$$WL_k = \sum_{i=1}^{N-1} \left| x_{i+1} - x_i \right|$$

7. **Slope sign changes:** it is similar to the zero crossings feature. It also provides information about the frequency content of the signal. It is calculated as follows Increment one unit the slope sign change value if:

$$\{x_i > x_{i-1} and\ x_i > x_{i+1}\}\ or\ \{x_i < x_{i-1} and\ x_i\ < x_{i+1}\}\ and\ \left|x_i - x_{i+1}\right| \geq\ threshold\ or\ \left|x_i - x_{i-1}\right| \geq threshold$$

Which can be rewritten as:

$$SSC_k = \sum_{i=2}^{N-1} [f[(x_i - x_{i-1})(x_i - x_{i+1})]];$$

$$f(x)\ =\ 1\ if\ x \geq threshold$$

$$f(x)\ =\ 0\ otherwise$$

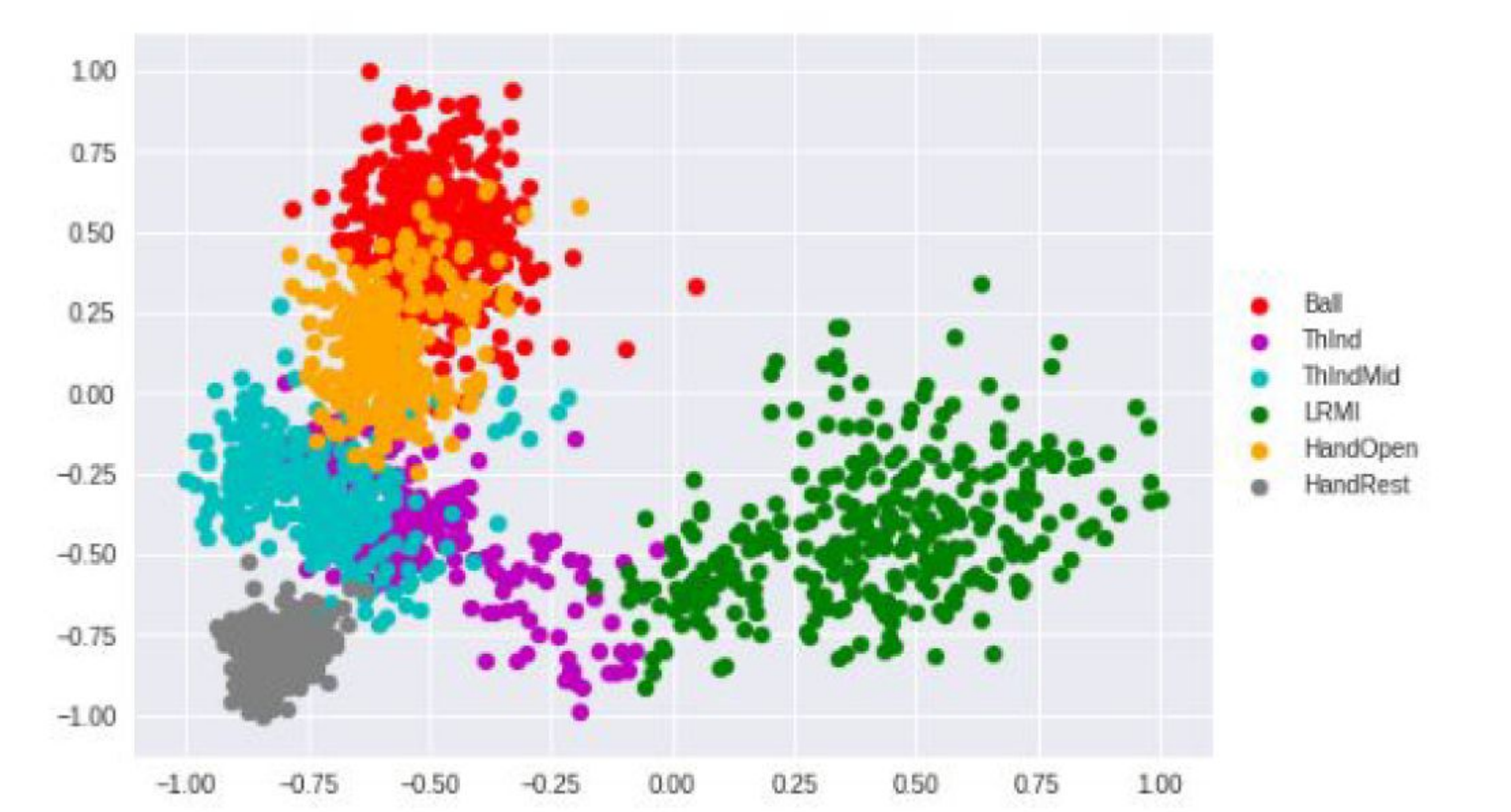
8. **Willison amplitude:** it is the number of times that the difference in amplitude between adjacent data points exceed a predefined threshold. This feature provides information about the muscle contraction level. It is defined as:

$$WAMP_k = \sum_{i=1}^{N-1} f(\left|x_i - x_{i+1}\right|)$$

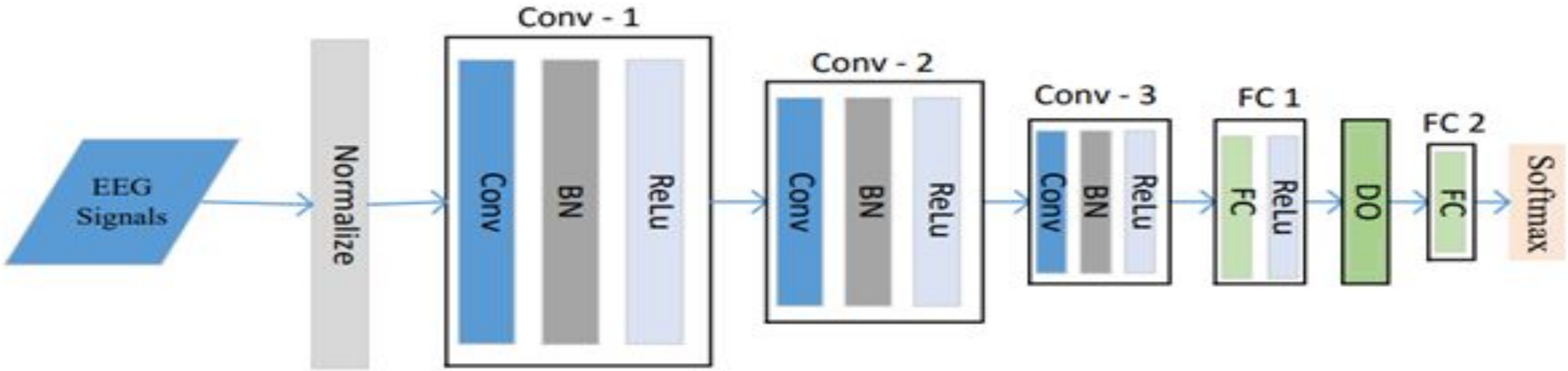
$$f(x)\ =\ 1\ if\ x\ \geq threshold$$

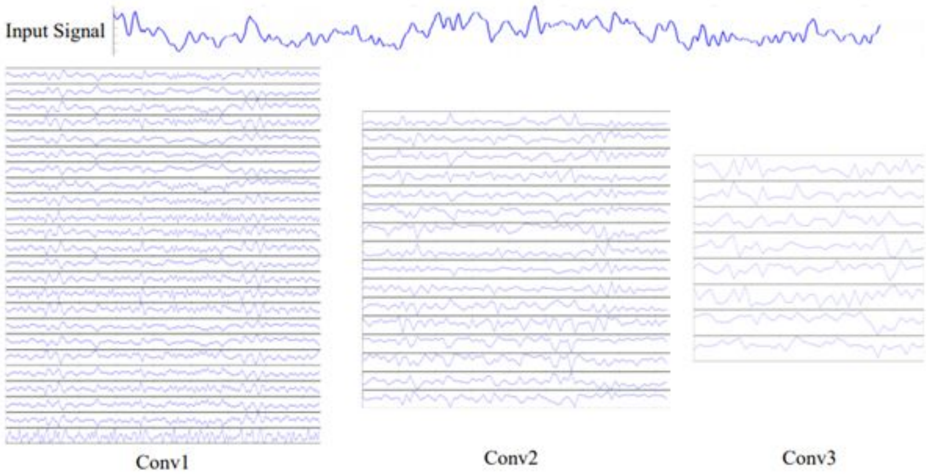
$$f(x)\ =\ 0\ otherwise$$

Résultat d'une segmentation + LDA



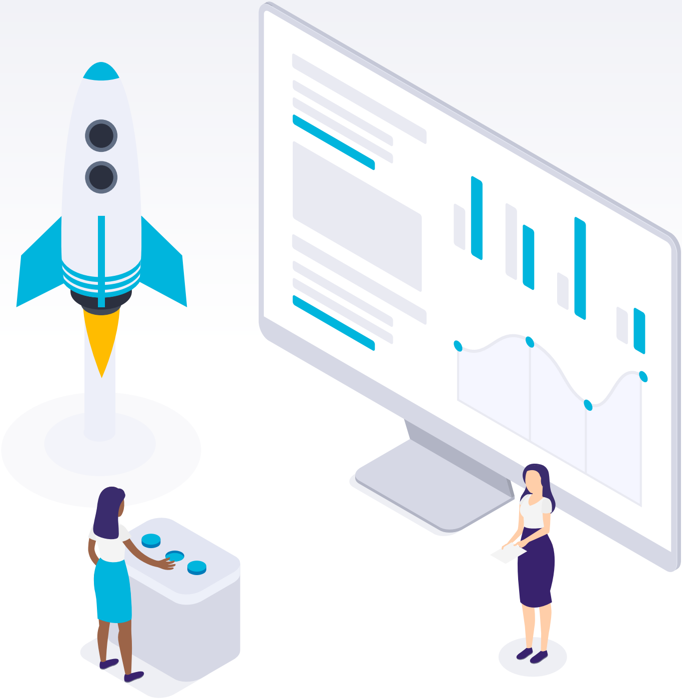
P-1D-CNN en utilisant le Deep Learning





4

Décomposition



Principe mathématique de la décomposition

Décomposition discrète:

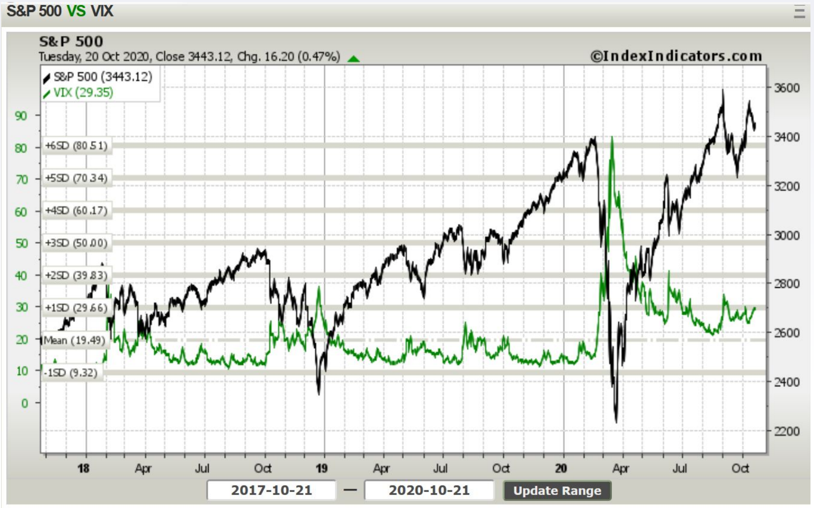
Un signal continu et périodique dans le temps peut être décomposé en une série des fonctions trigonométriques sinus et cosinus.

Ainsi, on parle d'une décomposition en Série de Fourier qui exprime le signal périodique comme **une combinaison linéaire** de signaux sinusoïdaux

$$x(t)=a_0+\sum_{n=1}^{+\infty} a_n \cos\left(n\frac{2\pi}{T}t\right)+b_n \sin\left(n\frac{2\pi}{T}t\right)$$

Somme de sinus et de cosinus : facile à interpréter

Principe mathématique de la décomposition



Problème :

Les séries temporelles financières sont continues mais non périodiques.

Solution pour les signaux continus et non périodiques

Décomposition continue:

$$X(f)=\int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}\,dt$$

Application avec un code en Python:

- **Signal sans bruit ayant une seule fréquence**
- **Signal sans bruit composé de deux fréquences**
- **Signal avec bruit composé de deux fréquences**

numpy.fft.fft

numpy.fft.fft(*a, n=None, axis=-1, norm=None*)

Compute the one-dimensional discrete Fourier Transform.

Meilleure performance avec numpy.fft.fft

numpy.fft.fft

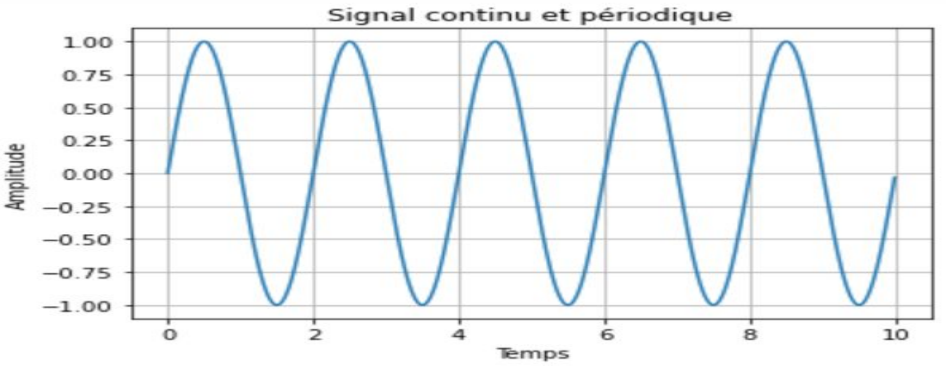
numpy.fft.fft(*a*, *n=None*, *axis=-1*, *norm=None*)

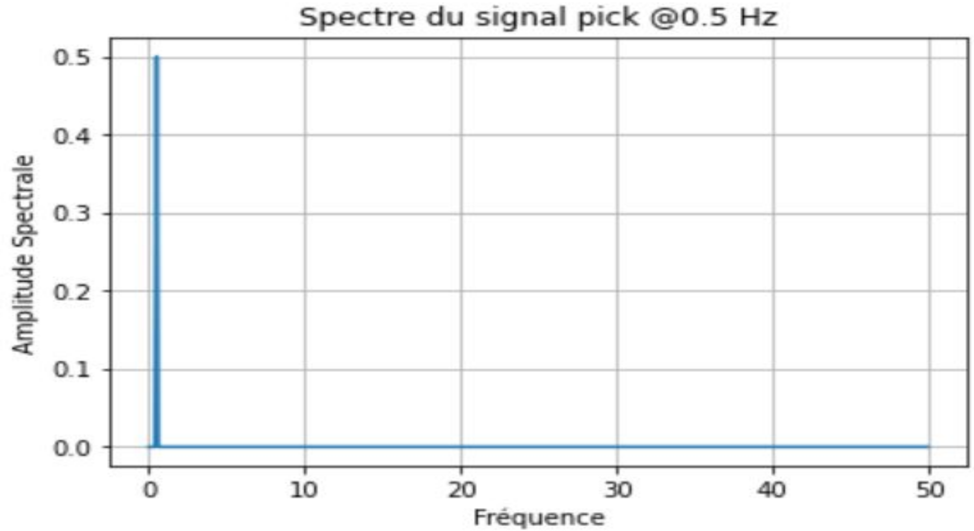
Compute the one-dimensional discrete Fourier Transform.

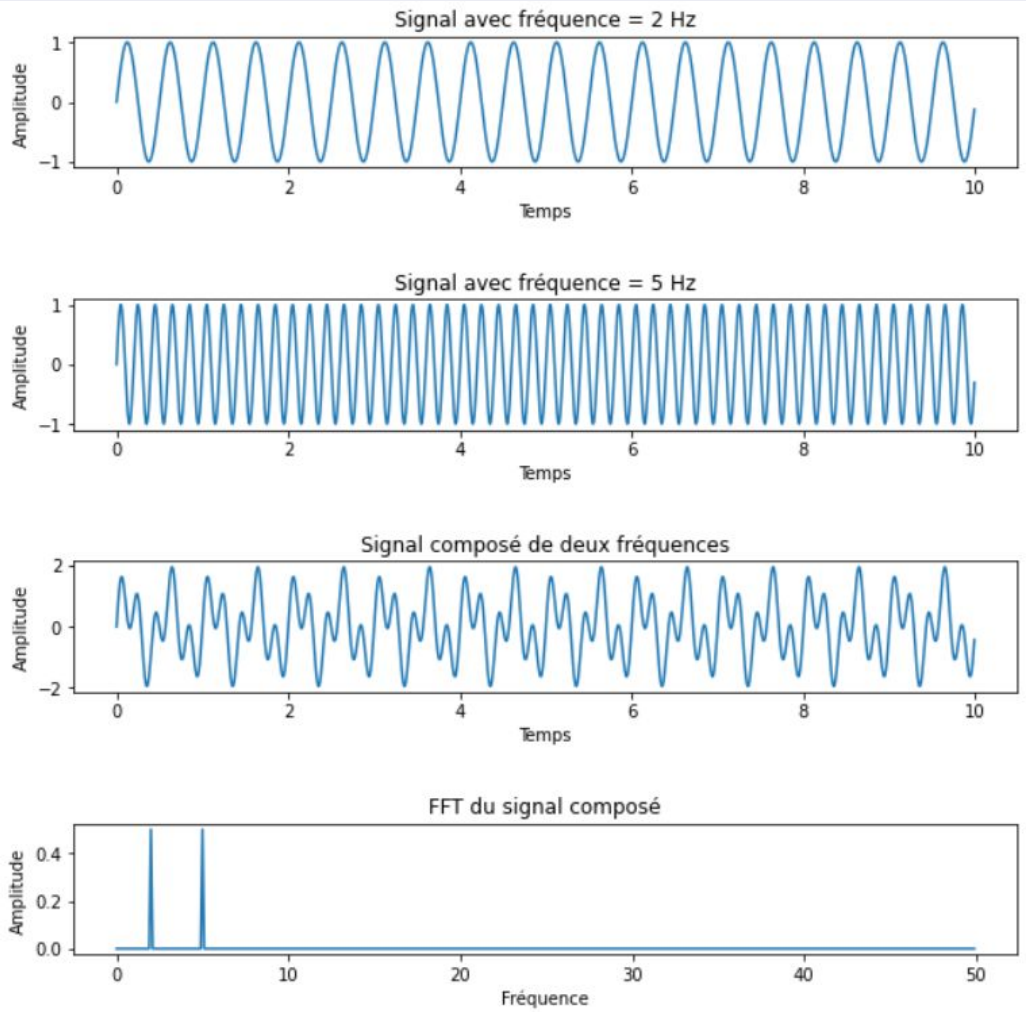
```
%timeit np.fft.fft(signal)
49.5 µs ± 3.17 µs per loop (mean ± std. dev. of 7 runs, 10000 loops each)

%timeit DFT_lent(signal)
969 ms ± 7.94 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

%timeit FFT(signal)
10.6 ms ± 425 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)

%timeit FFT_vectorized(signal)
862 µs ± 23.5 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

```python
# Chargement des modules
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
# Fréquence d'échantillonnage
freq_sampling = 100
# Intervalles d'échantillonnage
interval_sampling = 1 / freq_sampling
# Temps initial
ti = 0
# Temps final
tf = 10
# Fréquence du signal
freq = 0.5
# Intervalle temporel
time = np.arange(ti, tf, interval_sampling)
# Construction du signal
amplitude = np.sin(2 * np.pi * freq * time)
# Calcul de la transformation de Fourier
trans_fourier = np.fft.fft(amplitude)
# Normalisation de la transformation
trans_fourier = trans_fourier / len(amplitude)
# On supprime la fréquence d'échantillonnage
trans_fourier = trans_fourier[range( int(len(amplitude) / 2 )  )]
# Paramètres graphiques
compteur    = len(amplitude)
values       = np.arange(int(compteur/2))
periode_temporelle  = compteur / freq_sampling
frequences  = values/periode_temporelle
```

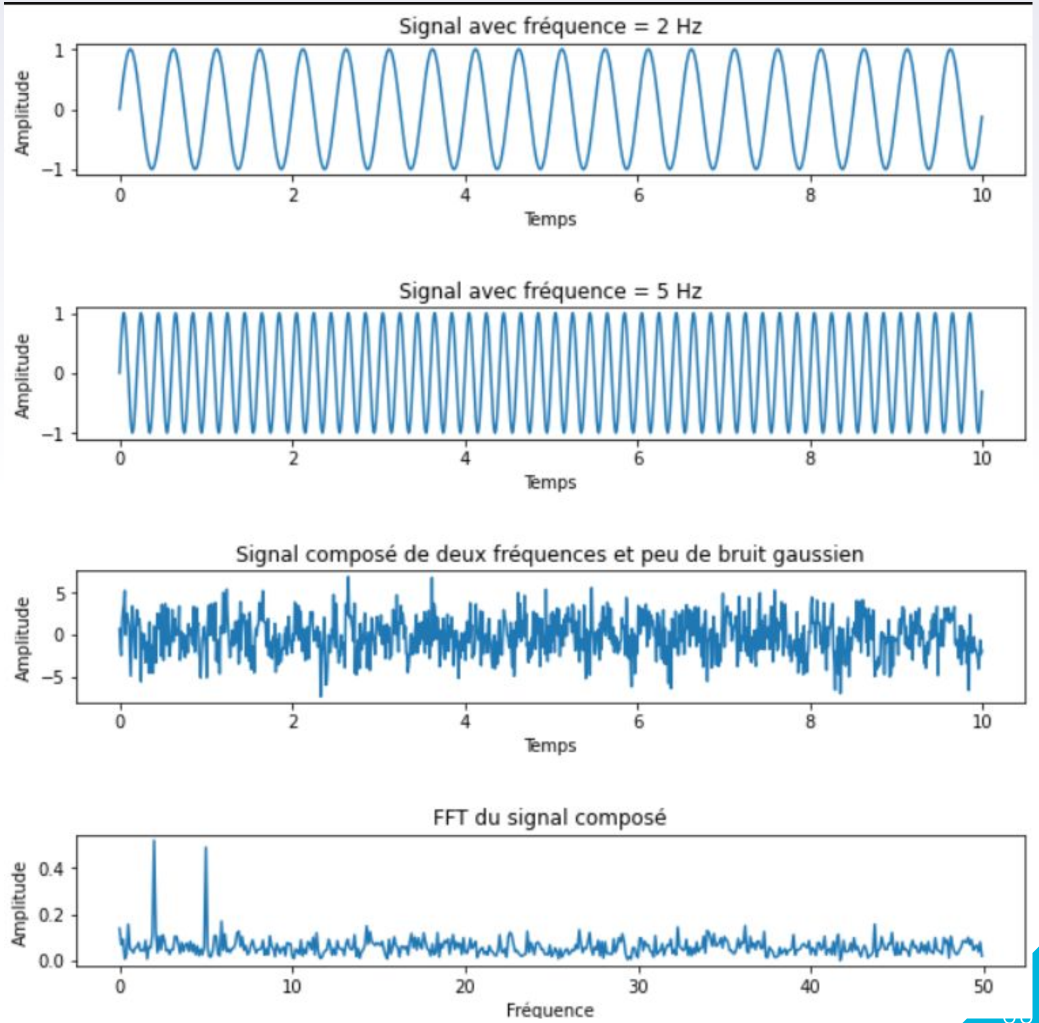
```python
plt.plot(time, amplitude)
plt.xlabel("Temps")
plt.ylabel("Amplitude")
plt.title("Signal continu et périodique")
plt.grid()
plt.show()
```



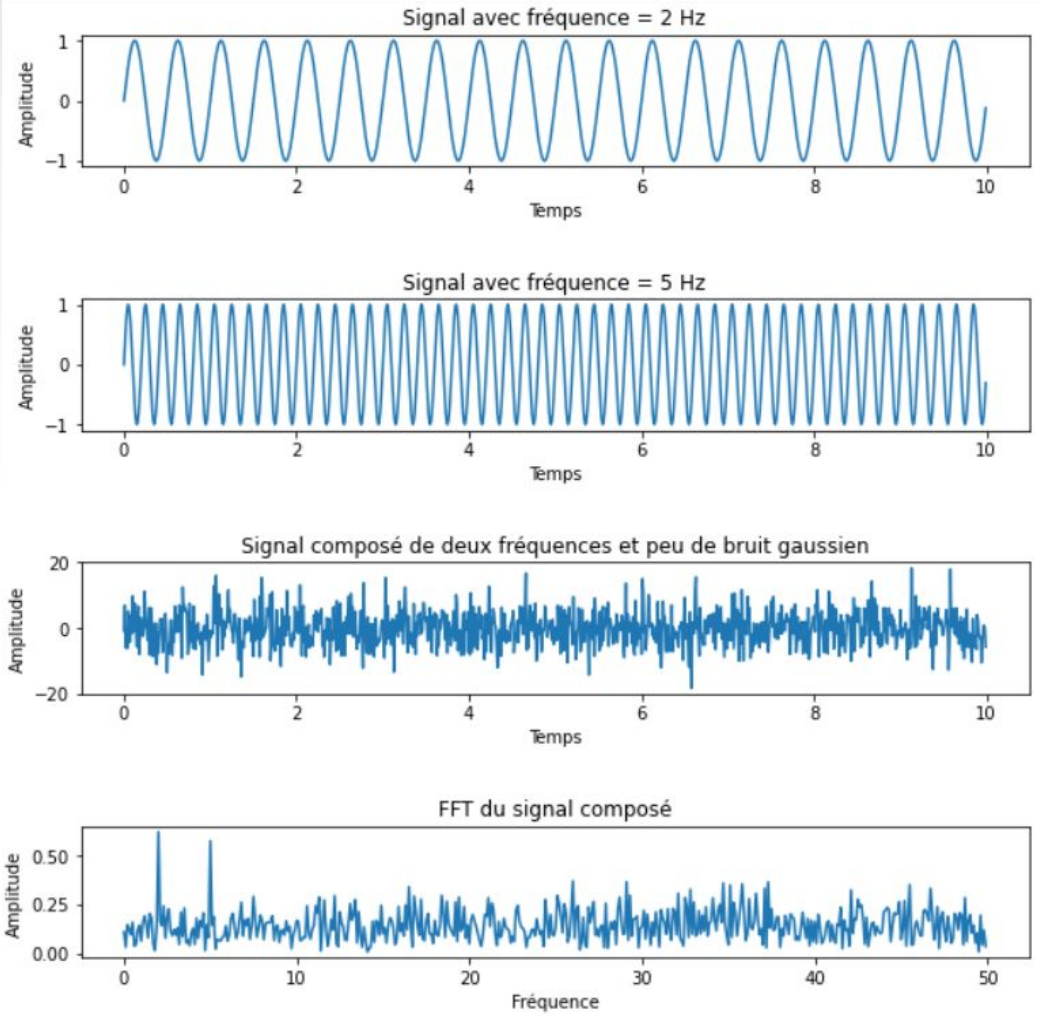




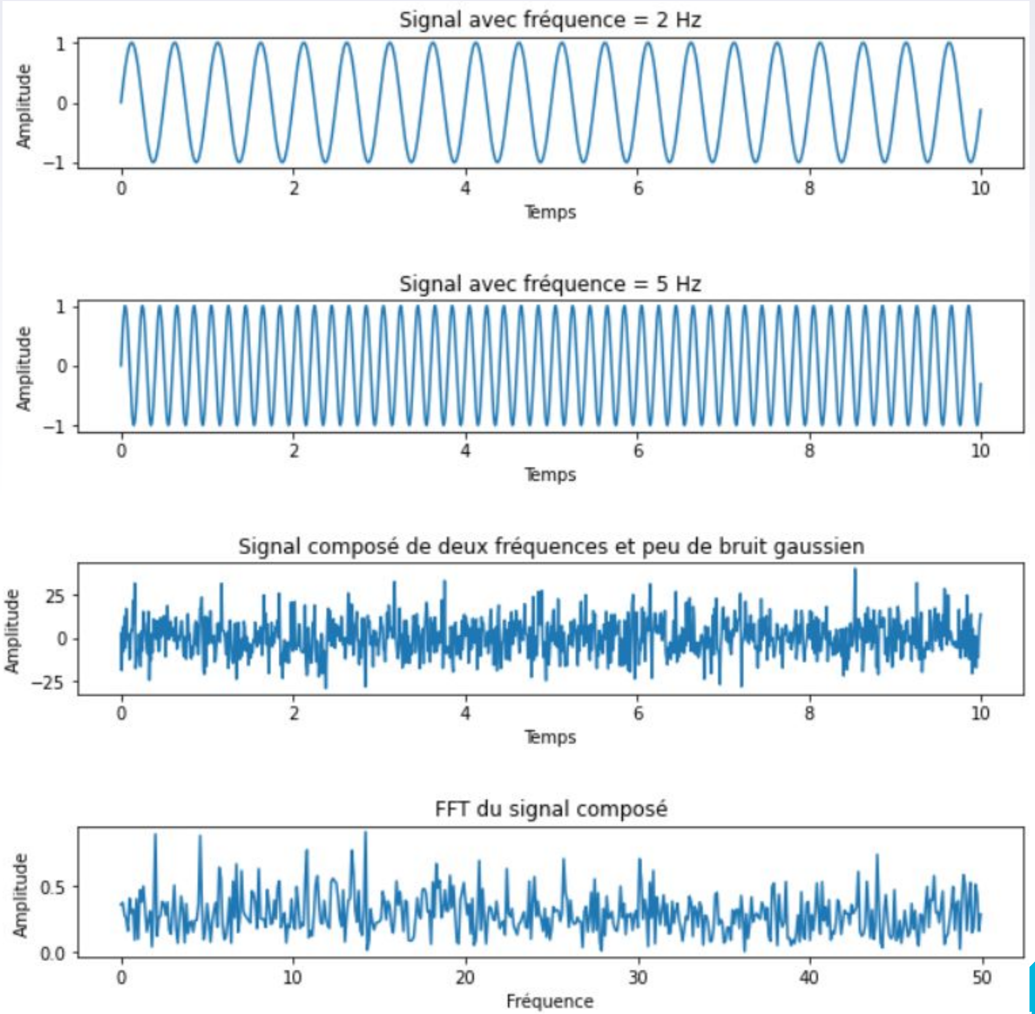
- La transformation de Fourier permet de détecter les deux fréquences composant le signal.



- Avec peu de bruit, la transformation de Fourier permet de détecter les deux fréquences composant le signal.



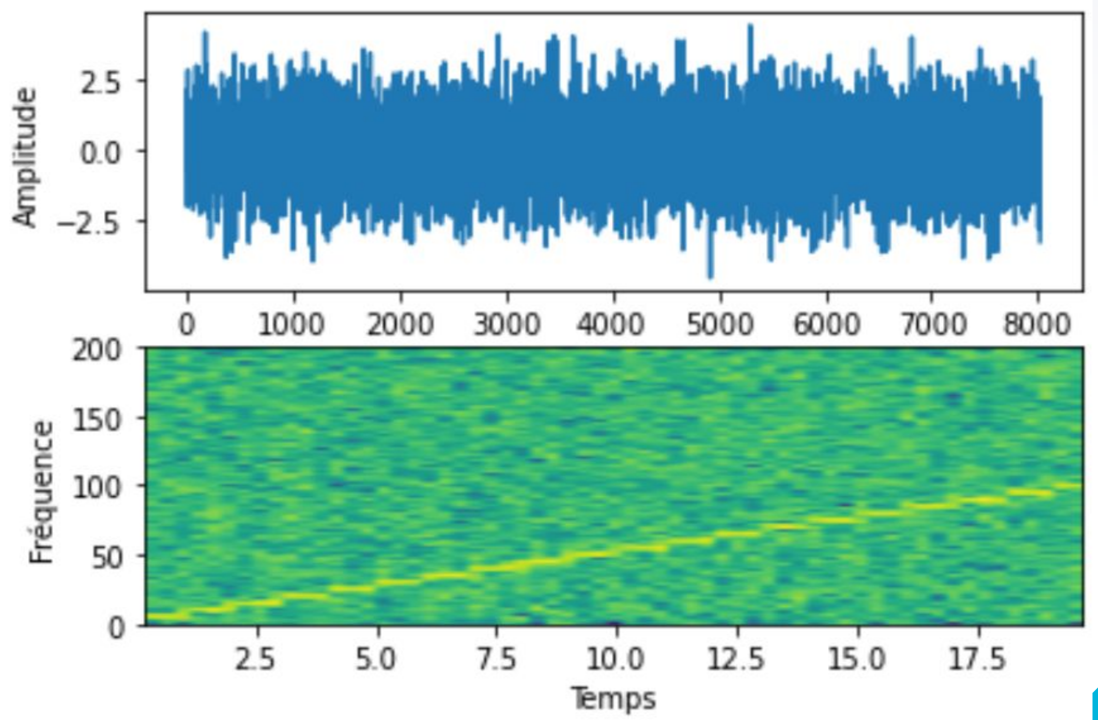
- Avec un bruit moyen, la transformation de Fourier arrive quand même à détecter les deux fréquences composant le signal.



- Avec beaucoup de bruit, il est presque impossible de détecter les deux fréquences

Interprétation à l'aide d'un spectrogramme

```
fig, (ax1, ax2) = plt.subplots(nrows=2)
ax1.plot(t, x)
Pxx, freqs, bins, im = ax2.specgram(x, NFFT=NFFT, Fs=Fs, noverlap=900)
# The `specgram` method returns 4 objects. They are:
# - Pxx: the periodogram
# - freqs: the frequency vector
# - bins: the centers of the time bins
# - im: the .image.AxesImage instance represe
plt.show()
```



by Dr Time and Brother Frequency

Integrate your function times a complex exponential It's really not so hard you can do it with your pencil And when you're done with this calculation You've got a brand new function - the Fourier Transformation

What a prism does to sunlight, what the ear does to sound

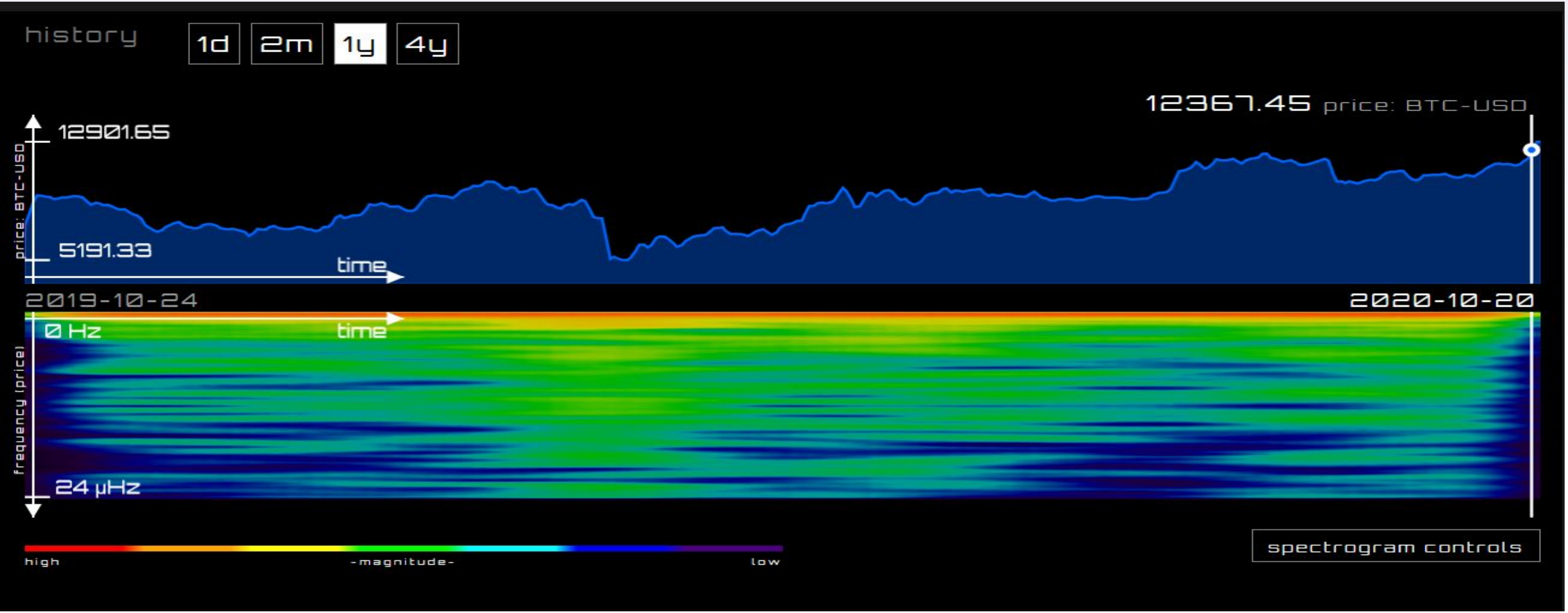
Fourier does to signals, it's the coolest trick around

Now filtering is easy, you don't need to convolve

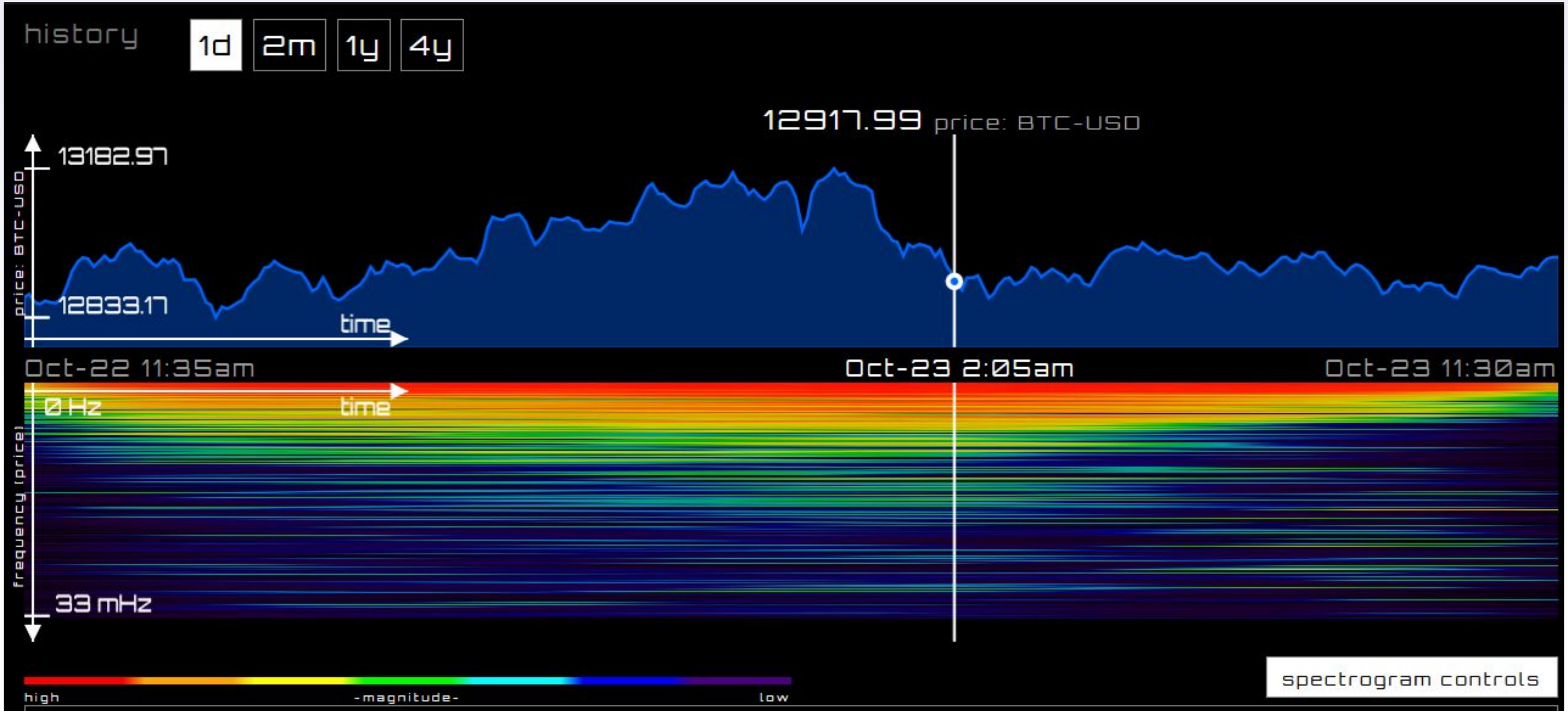
All you do is multiply in order to solve. From time into frequency - from frequency to time Every operation in the time domain Has a Fourier analog - that's what I claim Think of a delay, a simple shift in time It becomes a phase rotation - now that's truly sublime!And to differentiate, here's a simple trick Just multiply by J omega, ain't that slick?Integration is the inverse, what you gonna do?

Divide instead of multiply - you can do it too. From time into frequency - from frequency to time etc …

La suite + l'interprétation sur http://www.jmlg.org/lyrics/Fouriers_Song.htm

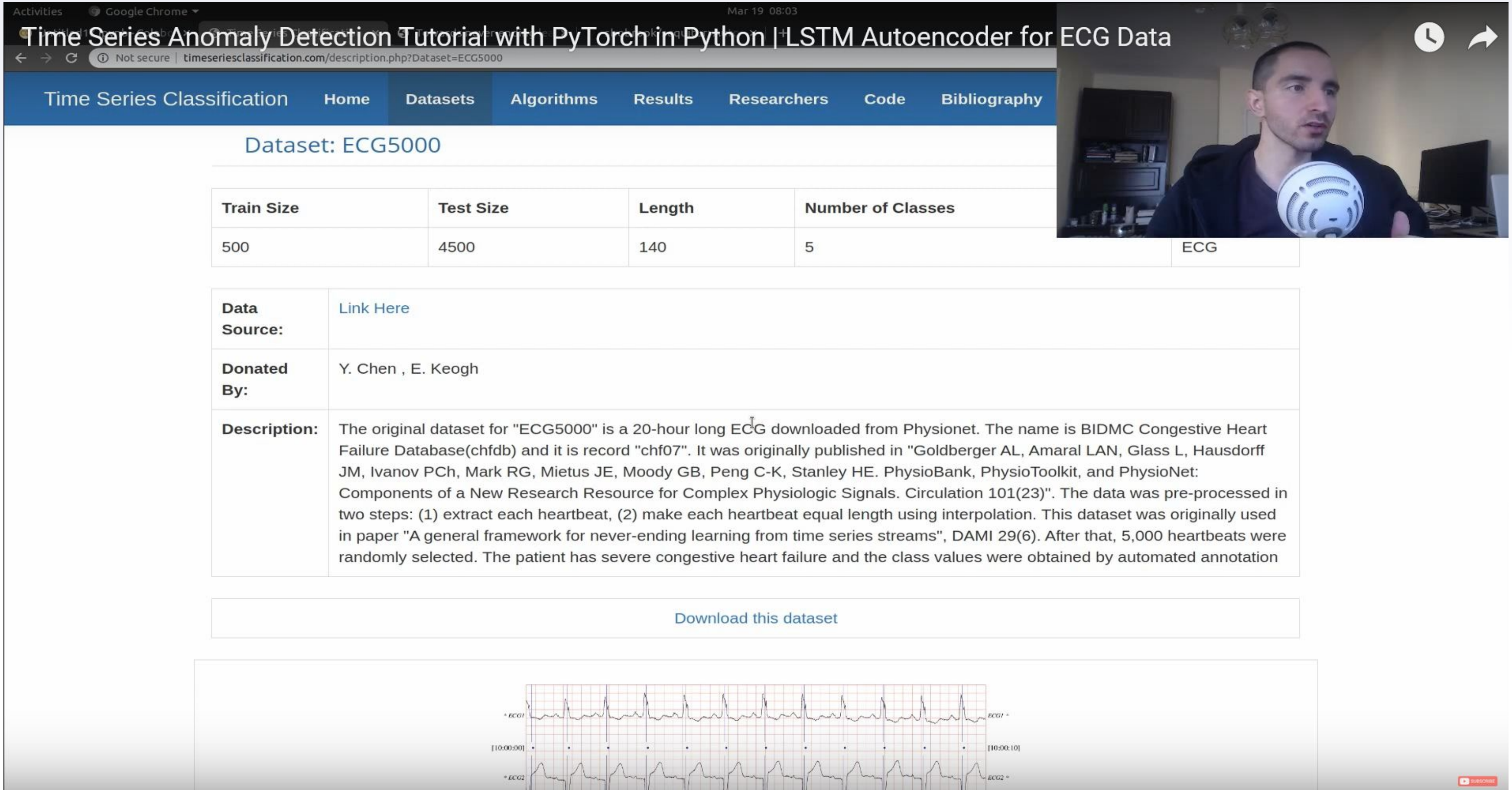


**** Quand les variations sont relativement stables ==> le spectre contient principalement relativement des faibles fréquences.
**** Quand les variations sont grandes (il y a de la volatilité) ==> tout le spectre s'illumine et ces événements ressemblent à des flammes.

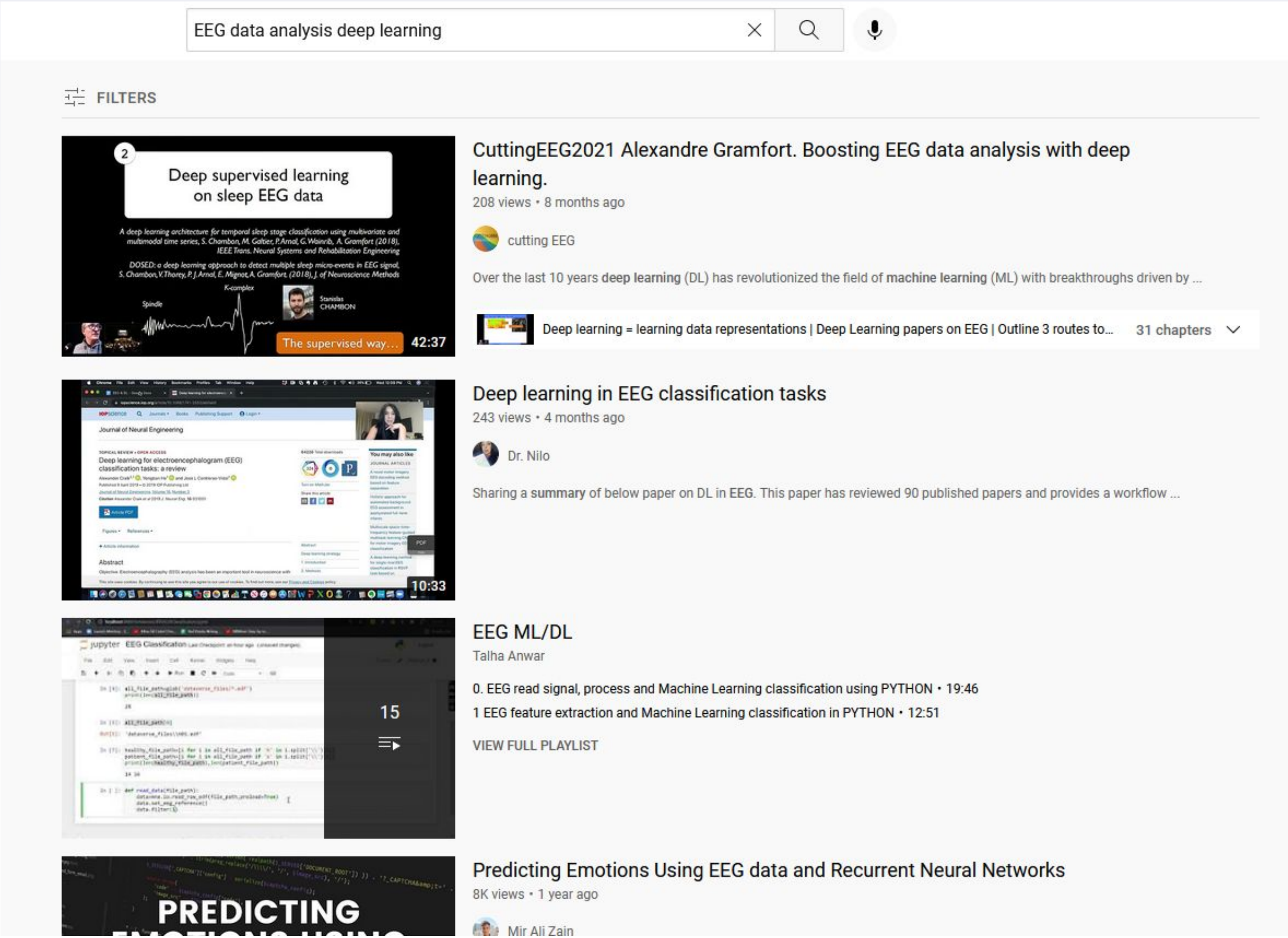


**** Quand les variations sont relativement stables ==> le spectre contient principalement relativement des faibles fréquences.
**** Quand les variations sont grandes (il y a de la volatilité) ==> tout le spectre s'illumine et ces événements ressemblent à des flammes.

Ressources


Activities
Google Chrome
Mar 19 08:03
Time Series Anomaly Detection Tutorial with PyTorch in Python | LSTM Autoencoder for ECG Data
Not secure | timeseriesclassification.com/description.php?Dataset=ECG5000
Time Series Classification
Home
Datasets
Algorithms
Results
Researchers
Code
Bibliography
Dataset: ECG5000
Train Size
Test Size
Length
Number of Classes
500
4500
140
5
ECG
Data Source:
Link Here
Donated By:
Y. Chen , E. Keogh
Description:
The original dataset for "ECG5000" is a 20-hour long ECG downloaded from Physionet. The name is BIDMC Congestive Heart Failure Database(chfdb) and it is record "chf07". It was originally published in "Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23)". The data was pre-processed in two steps: (1) extract each heartbeat, (2) make each heartbeat equal length using interpolation. This dataset was originally used in paper "A general framework for never-ending learning from time series streams", DAMI 29(6). After that, 5,000 heartbeats were randomly selected. The patient has severe congestive heart failure and the class values were obtained by automated annotation
Download this dataset
ECG1
ECG1
[10:00:00]
[10:00:10]
ECG2
ECG2

Ressources


EEG data analysis deep learning
FILTERS
Deep supervised learning on sleep EEG data
The supervised way...
42:37
CuttingEEG2021 Alexandre Gramfort. Boosting EEG data analysis with deep learning.
208 views • 8 months ago
cutting EEG
Over the last 10 years deep learning (DL) has revolutionized the field of machine learning (ML) with breakthroughs driven by ...
Deep learning = learning data representations | Deep Learning papers on EEG | Outline 3 routes to...
31 chapters
10:33
Deep learning in EEG classification tasks
243 views • 4 months ago
Dr. Nilo
Sharing a summary of below paper on DL in EEG. This paper has reviewed 90 published papers and provides a workflow ...
15
EEG ML/DL
Talha Anwar
0. EEG read signal, process and Machine Learning classification using PYTHON • 19:46
1 EEG feature extraction and Machine Learning classification in PYTHON • 12:51
VIEW FULL PLAYLIST
PREDICTING
Predicting Emotions Using EEG data and Recurrent Neural Networks
8K views • 1 year ago
Mir Ali Zain

Ressources

Browse State-of-the-Art | Datasets | Methods | More

EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces

23 Nov 2016 · Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, Brent J. Lance · Edit social preview

Brain computer interfaces (BCI) enable direct communication with a computer, using neural activity as the control signal. This neural signal is generally chosen from a variety of well-studied electroencephalogram (EEG) signals. For a given BCI paradigm, feature extractors and classifiers are tailored to the distinct characteristics of its expected EEG control signal, limiting its application to that specific signal. Convolutional Neural Networks (CNNs), which have been used in computer vision and speech recognition, have successfully been applied to EEG-based BCIs; however, they have mainly been applied to single BCI paradigms and thus it remains unclear how these architectures generalize to other paradigms. Here, we ask if we can design a single CNN architecture to accurately classify EEG signals from different BCI paradigms, while simultaneously being as compact as possible. In this work we introduce EEGNet, a compact convolutional network for EEG-based BCIs. We introduce the use of depthwise and separable convolutions to construct an EEG-specific model which encapsulates well-known EEG feature extraction concepts for BCI. We compare EEGNet to current state-of-the-art approaches across four BCI paradigms: P300 visual-evoked potentials, error-related negativity responses (ERN), movement-related cortical potentials (MRCP), and sensory motor rhythms (SMR). We show that EEGNet generalizes across paradigms better than the reference algorithms when only limited training data is available. We demonstrate three different approaches to visualize the contents of a trained EEGNet model to enable interpretation of the learned features. Our results suggest that EEGNet is robust enough to learn a wide variety of interpretable features over a range of BCI tasks, suggesting that the observed performances were not due to artifact or noise sources in the data.

PDF | Abstract

Code

Edit

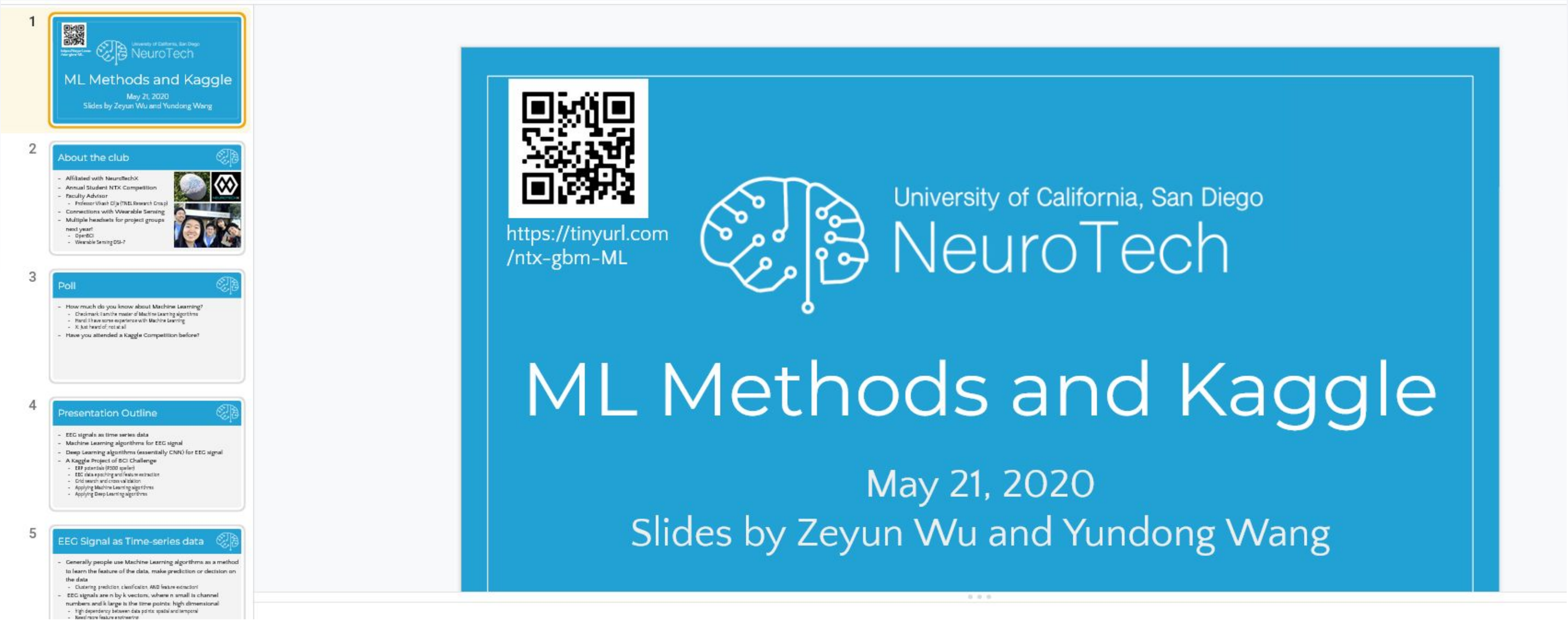
vlawhern/arl-eegmodels official	★ 666	TensorFlow
aliasvishnu/EEGNet	★ 137	PyTorch
gibranfp/P300-CNNT	★ 41	TensorFlow

Tasks

Edit

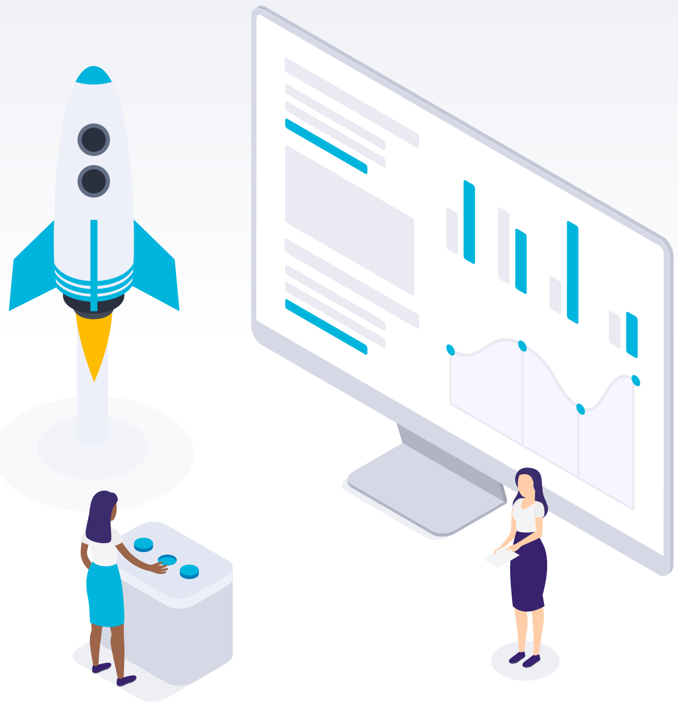
Computer Vision | EEG | speech-recognition

Ressources



1
ML Methods and Kaggle
May 21, 2020
Slides by Zeyun Wu and Yundong Wang
2
About the club
- Affiliated with NeuroTechX
- Annual Student NTX Competition
- Faculty Advisor
- Connections with Wearable Sensing
- Multiple headsets for project groups next year!
3
Poll
- How much do you know about Machine Learning?
- Have you attended a Kaggle Competition before?
4
Presentation Outline
- EEG signals as time series data
- Machine Learning algorithms for EEG signal
- Deep Learning algorithms (essentially CNN) for EEG signal
- A Kaggle Project of BCI Challenge
5
EEG Signal as Time-series data
- Generally people use Machine Learning algorithms as a method to learn the feature of the data, make prediction or decision on the data
- EEG signals are n by k vectors, where n small is channel numbers and k large is the time points: high dimensional
https://tinyurl.com
/ntx-gbm-ML
University of California, San Diego
NeuroTech
ML Methods and Kaggle
May 21, 2020
Slides by Zeyun Wu and Yundong Wang

4 Conclusion



Measure	Description
	Statistical measures
MeanRR	Mean of RRIs
RMSSD	Square root of the mean squared differences of successive RRIs
VAR	Variance of RRIs
SDNN	Standard deviation of RRIs
SDSD	Standard deviation of differences between adjacent RRIs
NN50	Number of intervals differences of successive NN intervals greater than 50ms
pNN50	Proportion of NN50 to the total number of NN intervals
NN20	Number of intervals differences of successive NN intervals greater than 20ms
pNN20	Proportion of NN20 to the total number of NN intervals
	Geometric measures
HRV Triangular Index	Total number of NN intervals divided by the height of the histogram of all NN intervals.
TINN	Baseline width of the minimal squared differences of the interpolation of the highest peak I the histogram of all NN intervals.

Measure	Description
VLF power	Power of the very low frequency band ($\leq$ 0.04Hz) of the PSD.
LF Power	Power of the low frequency band (0.04 – 0.15 Hz) of the PSD.
HF power	Power of the high frequency band (0.15 – 0.4 Hz) of the PSD.
LF/HF	Ratio of LF to HF.