Real Time Classifier for transient signals in Gravitational Waves

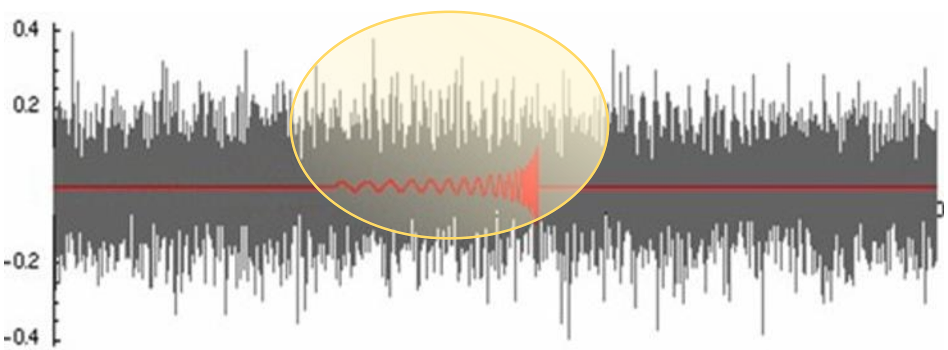
From raw data to classified triggers

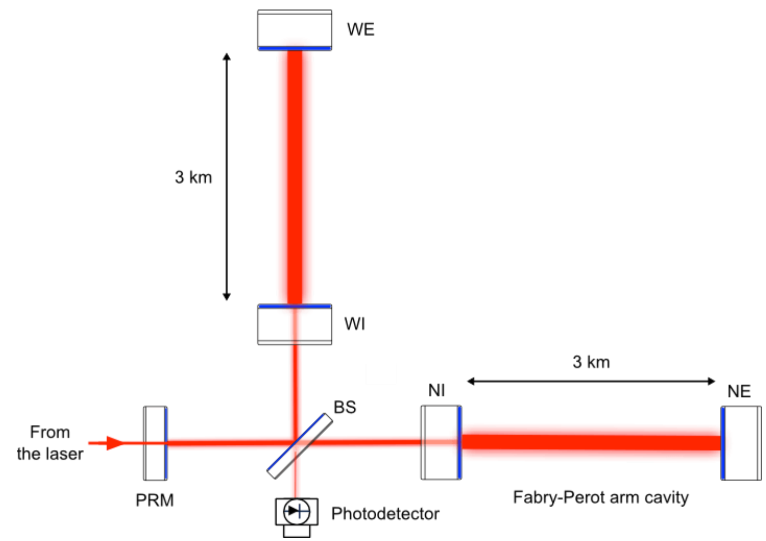
Elena Cuoco

European Gravitational Observatory and Scuola Normale Superiore

LIGO/Virgo data

- are time series sequences… **noisy time series** with low amplitude GW signal buried in





Which kind of astrophysical source?

- Known waveform
- Transient signal

Compact binary coalescence (CBC)

- Unknown waveform
- Transient signal

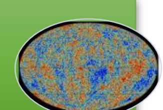
Core Collapse Supernovae (CCSN)

- Known waveform
- Persistent signal

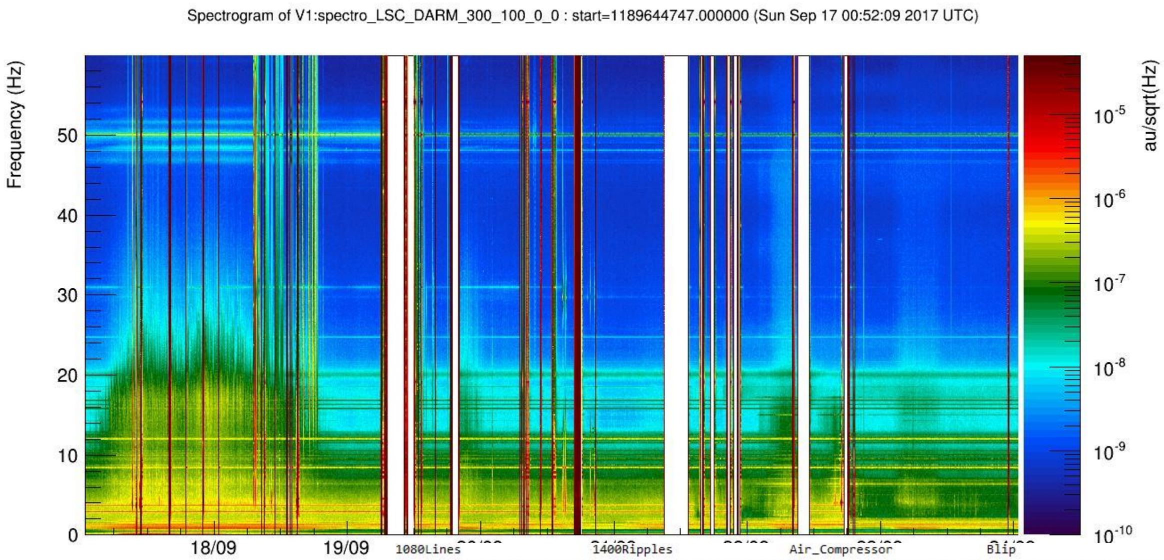
Continuous Waves (CW)

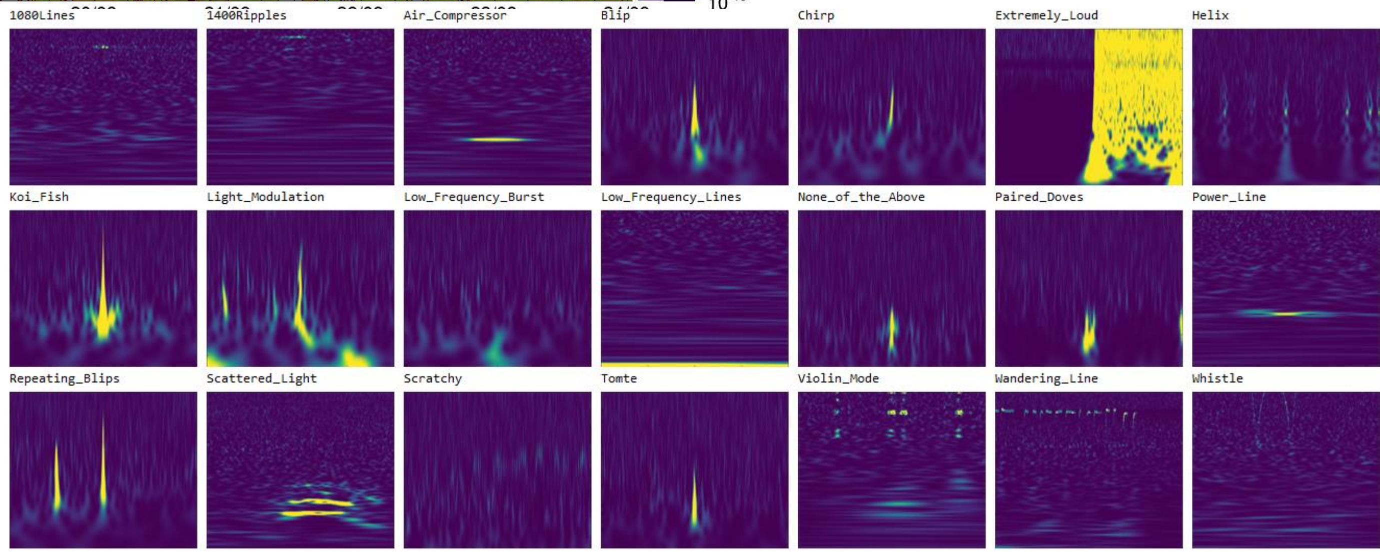
- Unknown model
- Persistent signal

Stochastic Background (SB)



Which kind of noise?





Which kind of data analysis techniques?

Known transient signals

- Modeled search
  - Matched filter

Unknown transient signals

- Un-modeled search:
  - Excess of energy detector

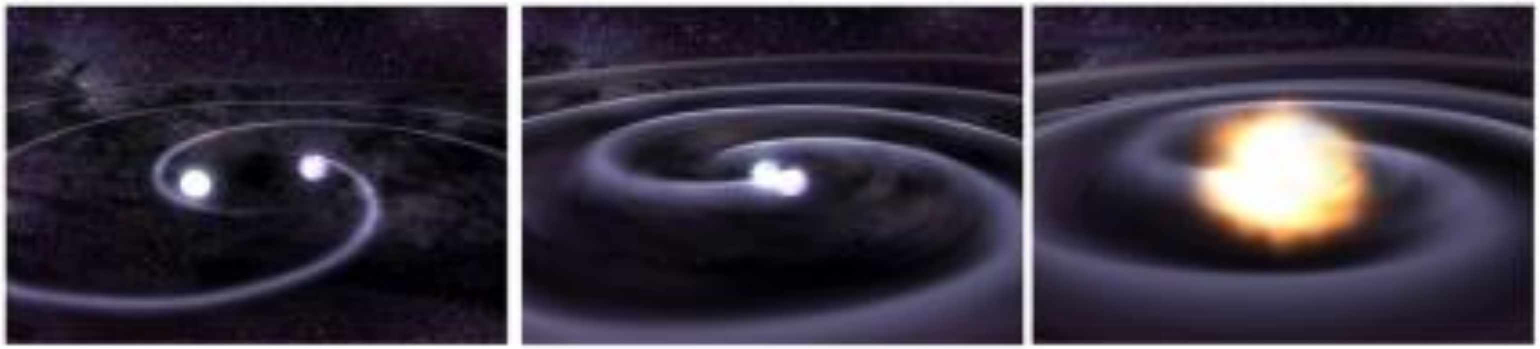
Known Continuous signals

- Modeled search
  - Integration method over long period

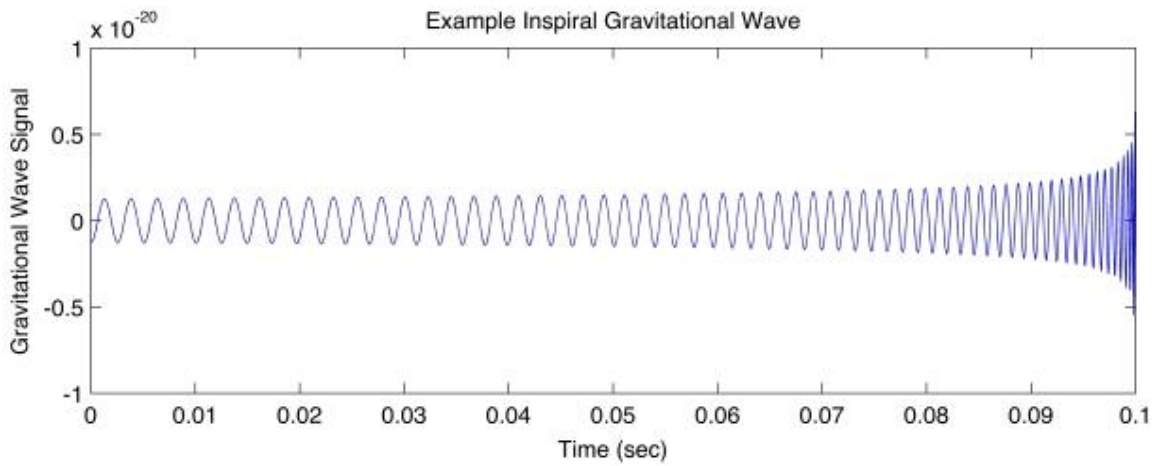
Unknown Stochastic signals

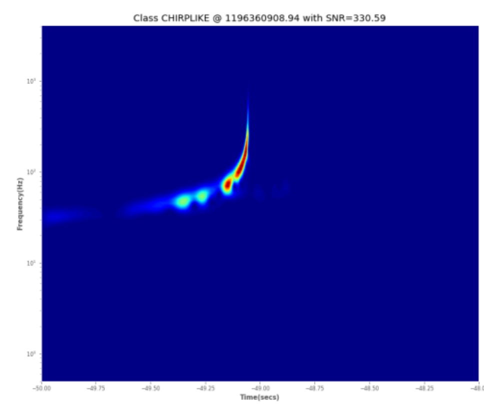
- Un-modeled search
  - Correlation between detectors

CBC Gravitational Wave signals



An artist's impression of two stars orbiting each other and progressing (from left to right) to merger with resulting gravitational waves. [Image: NASA/CXC/GSFC/T.Strohmayer]





An example signal from an inspiral gravitational wave source. [Image: A. Stuver/LIGO]

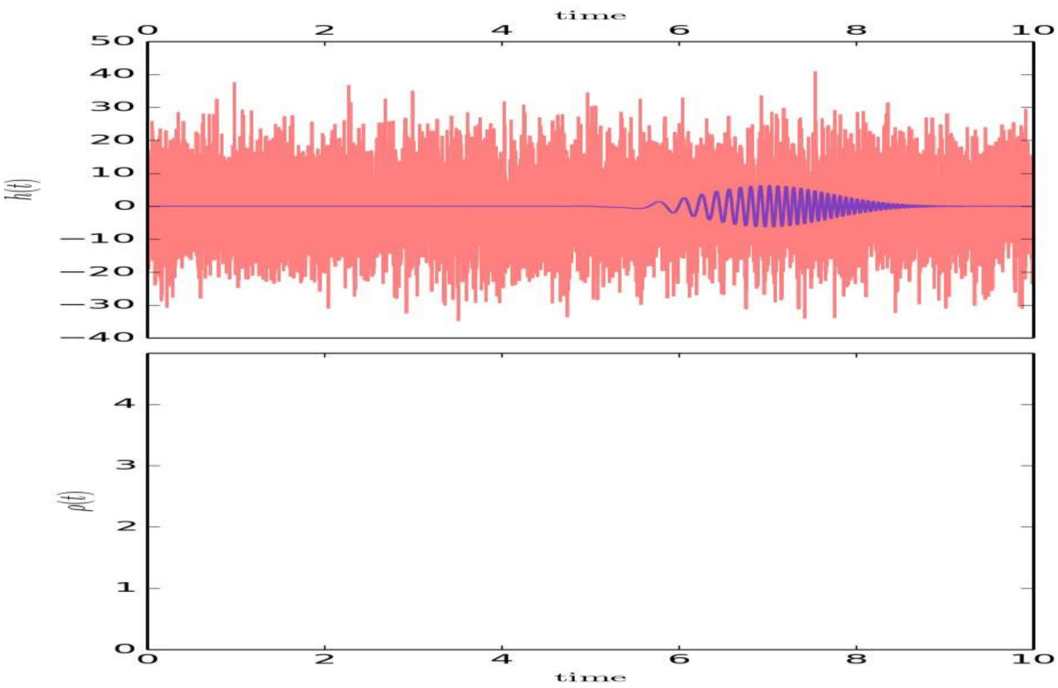
Modeled signals: Matched filter in action

Data

Template

$$\rho(t) = 4\int_0^\infty \frac{\tilde{x}(f)\ \tilde{h}^*(f)}{S_n(f)}\ e^{2\pi i f t}\ df$$

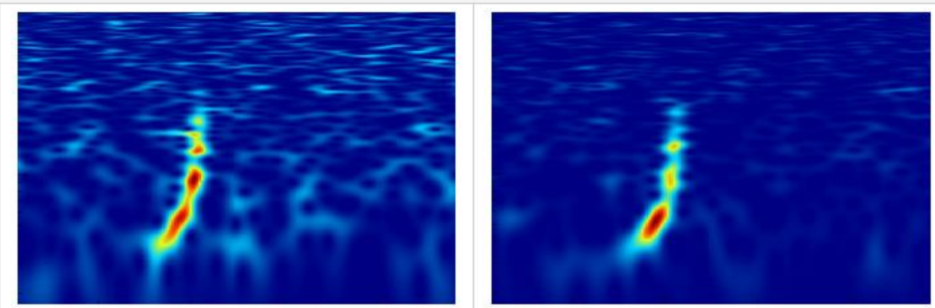
Noise power spectral density



Unmodeled signals

- Strategy: look for excess power in single detector or coherent/coincident in network data
- Example cWB (https://gwburst.gitlab.io/)

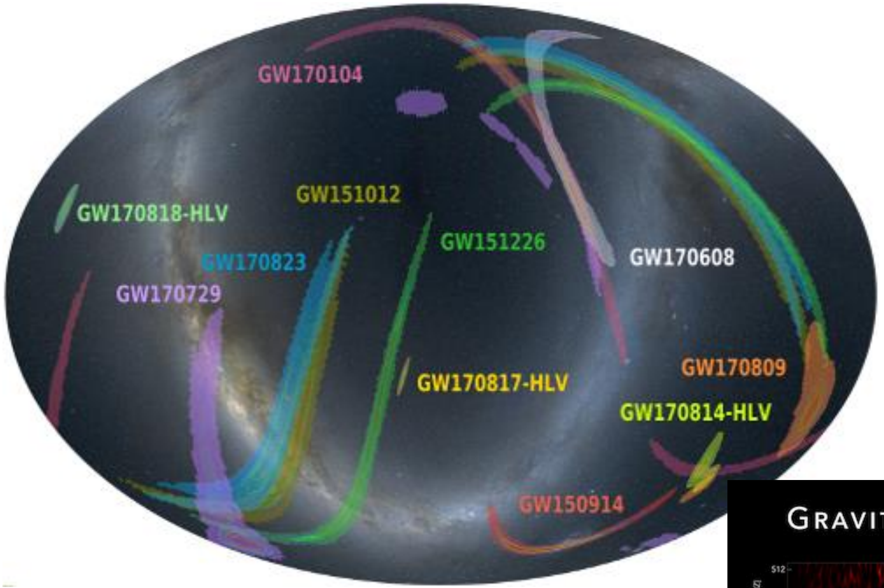
Coherent WaveBurst was used in the first direct detection of gravitational waves (GW150914) by LIGO and is used in the ongoing analyses on LIGO and Virgo data.



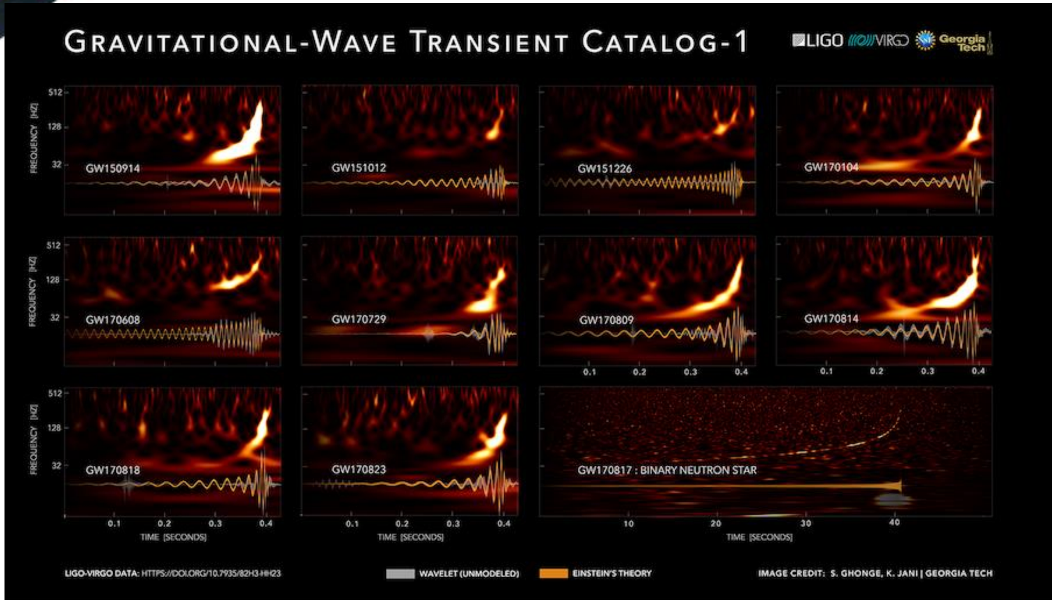
Time-Frequency maps of GW150914: Livingston data (left), Hanford data (right)
First screenshot of GW150914 event

arXiv:1811.12907

The first GW catalog (O1/O2 run)



[GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs]() arxiv.org/abs/1811.12907



O3 event rate ~1/week

GraceDB — Gravitational Wave Candidate Event Database

HOME | SEARCH | LATEST | DOCUMENTATION | LOGIN

Latest — as of 8 July 2019 13:15:27 UTC

Test and MDC events and superevents are not included in the search results by default; see the query help for information on how to search for events and superevents in those categories.

Query:

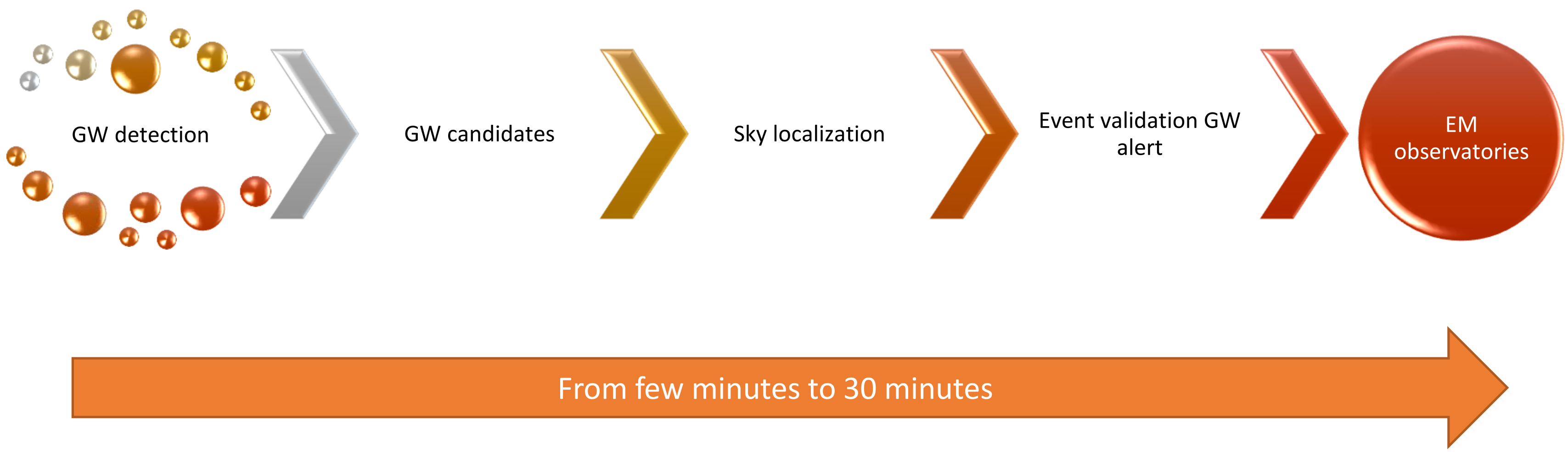
Search for: Superevent

Search

UID	Labels	t_start	t_0	t_end	FAR (Hz)	UTC Created
S190707q	ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1246527223.118398	1246527224.181226	1246527225.284180	5.265e-12	2019-07-07 09:33:44 UTC
S190706ai	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1246487218.321541	1246487219.344727	1246487220.585938	1.901e-09	2019-07-06 22:26:57 UTC
S190701ah	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1246048403.576563	1246048404.577637	1246048405.814941	1.916e-08	2019-07-01 20:33:24 UTC
S190630ag	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1245955942.175325	1245955943.179550	1245955944.183184	1.435e-13	2019-06-30 18:52:28 UTC
S190602aq	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1243533584.081266	1243533585.089355	1243533586.346191	1.901e-09	2019-06-02 17:59:51 UTC
S190524q	ADVNO SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1242708743.678669	1242708744.678669	1242708746.133301	6.971e-09	2019-05-24 04:52:30 UTC
S190521r	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1242459856.453418	1242459857.460739	1242459858.642090	3.168e-10	2019-05-21 07:44:22 UTC
S190521g	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1242442966.447266	1242442967.606934	1242442968.888184	3.801e-09	2019-05-21 03:02:49 UTC
S190519bj	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1242315361.378873	1242315362.655762	1242315363.676270	5.702e-09	2019-05-19 15:36:04 UTC
S190518bb	ADVNO SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1242242376.474609	1242242377.474609	1242242380.922655	1.004e-08	2019-05-18 19:19:39 UTC
S190517h	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1242107478.819517	1242107479.994141	1242107480.994141	2.373e-09	2019-05-17 05:51:23 UTC
S190513bm	ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1241816085.736106	1241816086.869141	1241816087.869141	3.734e-13	2019-05-13 20:54:48 UTC
S190512at	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1241719651.411441	1241719652.416286	1241719653.518066	1.901e-09	2019-05-12 18:07:42 UTC
S190510g	ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1241492396.291636	1241492397.291636	1241492398.293185	8.834e-09	2019-05-10 03:00:03 UTC
S190503bf	ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1240944861.288574	1240944862.412598	1240944863.422852	1.636e-09	2019-05-03 18:54:26 UTC
S190426c	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1240327332.331668	1240327333.348145	1240327334.353516	1.947e-08	2019-04-26 15:22:15 UTC
S190425z	ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK	1240215502.011549	1240215503.011549	1240215504.018242	4.538e-13	2019-04-25 08:18:26 UTC
S190421ar	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1239917953.250977	1239917954.409180	1239917955.409180	1.489e-08	2019-04-21 21:39:16 UTC
S190412m	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1239082261.146717	1239082262.222168	1239082263.229492	1.683e-27	2019-04-12 05:31:03 UTC
S190408an	PE_READY ADVOK SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK GCN_PRELIM_SENT	1238782699.268296	1238782700.287958	1238782701.359863	2.811e-18	2019-04-08 18:18:27 UTC
S190405ar	ADVNO SKYMAP_READY EMBRIGHT_READY PASTRO_READY DQOK	1238515307.863646	1238515308.863646	1238515309.863646	2.141e-04	2019-04-05 16:01:56 UTC

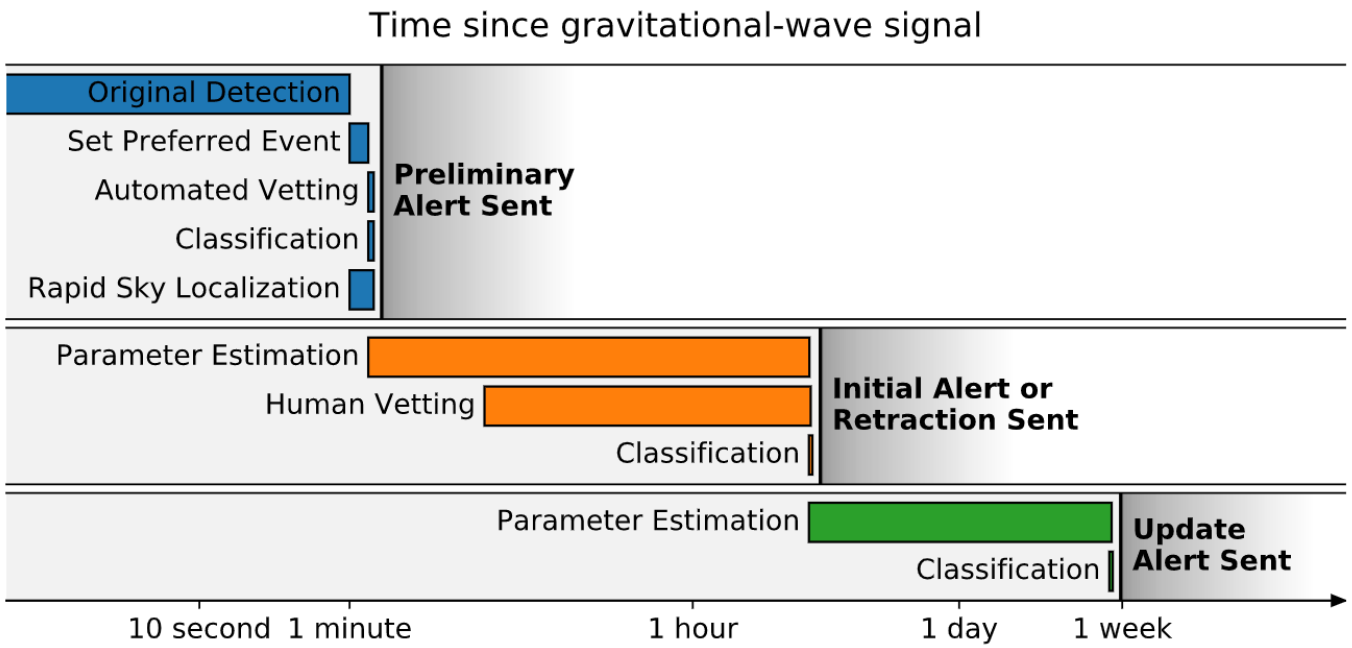
How Machine Learning can help real time analysis

Low Latency data analysis



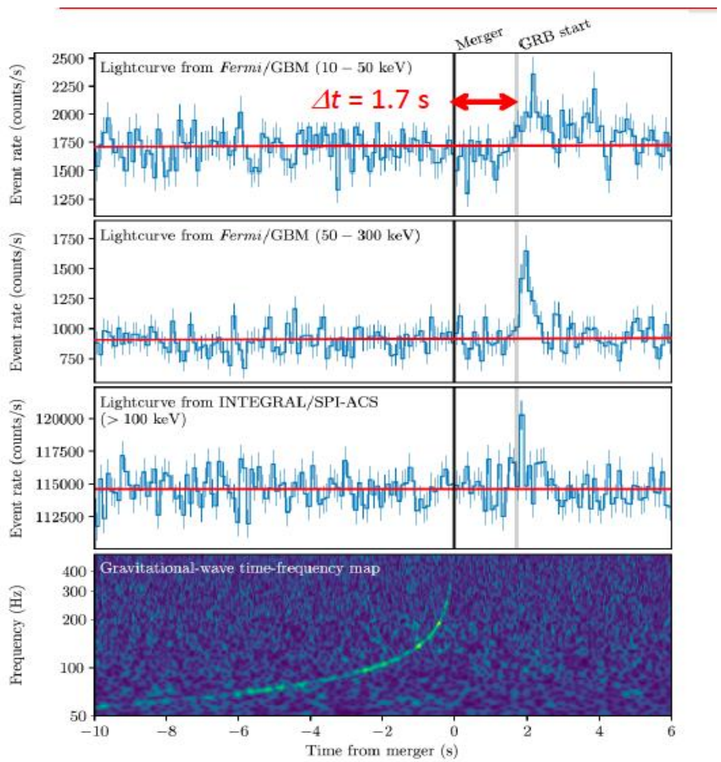
EGO EUROPEAN GRAVITATIONAL OBSERVATORY

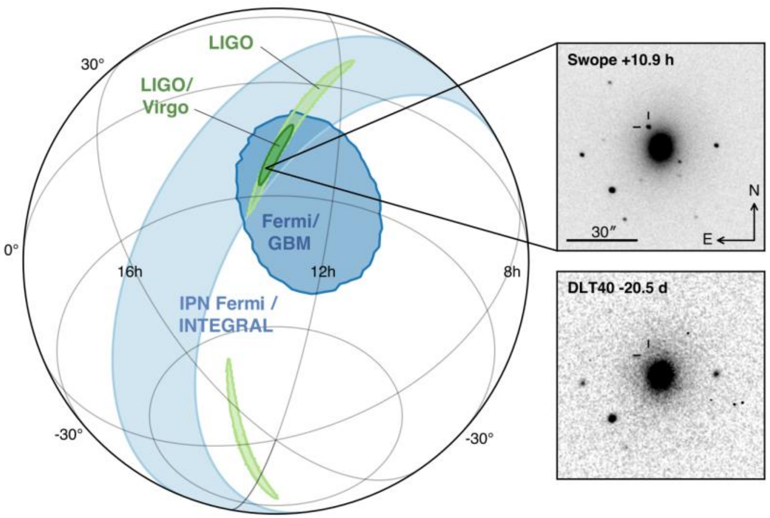
GW alert system



https://emfollow.docs.ligo.org/userguide/index.html

17 August 2017, 12:41:04 UT: The MultiMessenger Astronomy





DOI:10.1103/PhysRevLett.119.161101.

Numbers about Virgo data

Data Stream Flux	Data on disk	Number of events	Number of glitches
• 50MB/s	• 1-3PB	• 1/week • 1/day?	• 1/sec • 0.1/sec?

Should be analysed in less than 1min

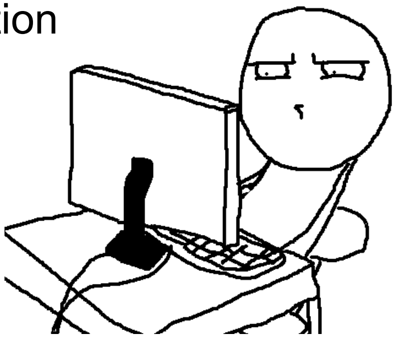
How Machine Learning can help

Data conditioning

- Identify Non linear noise coupling
- Use Deep Learning to remove noise
- Extract useful features to clean data

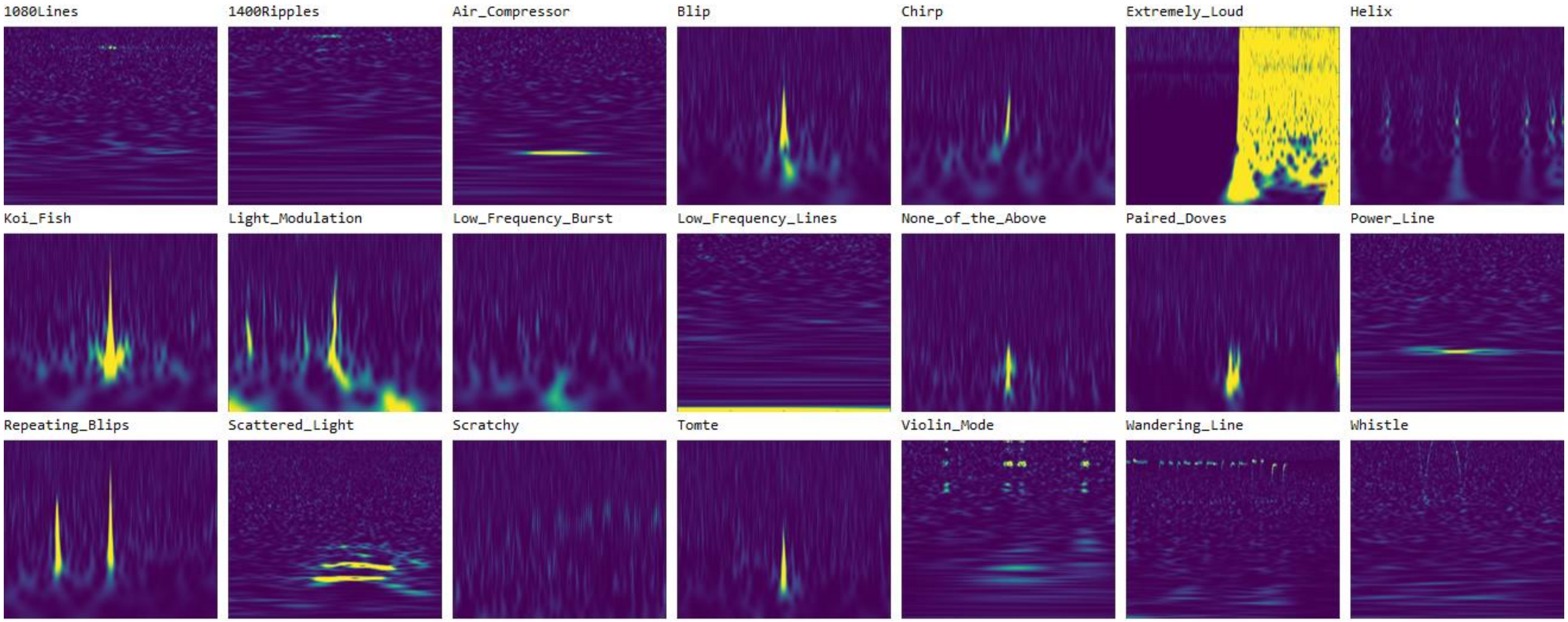
Signal Detection/Classification/PE

- A lot of fake signals due to noise
- Fast alert system
- Manage parameter estimation



Example of Glitch signals

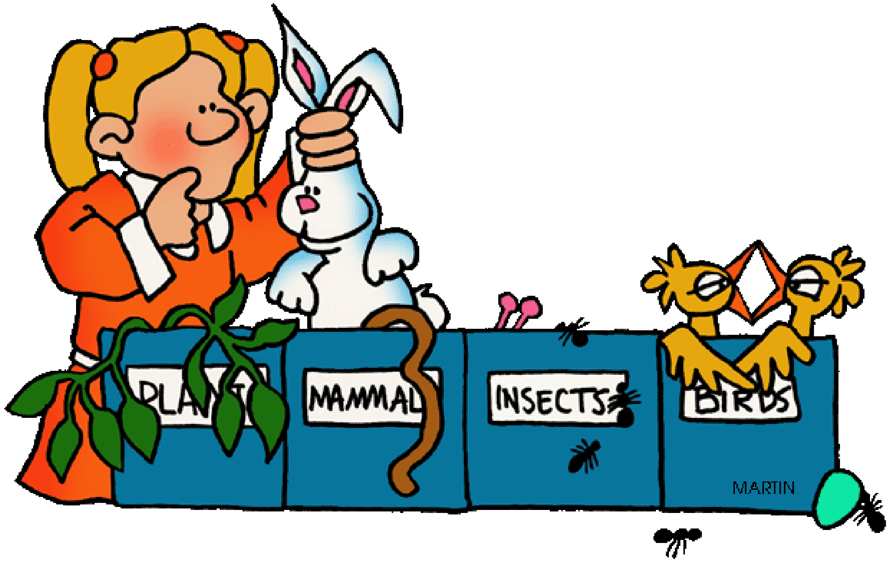
https://www.zooniverse.org/projects/zooniverse/gravity-spy



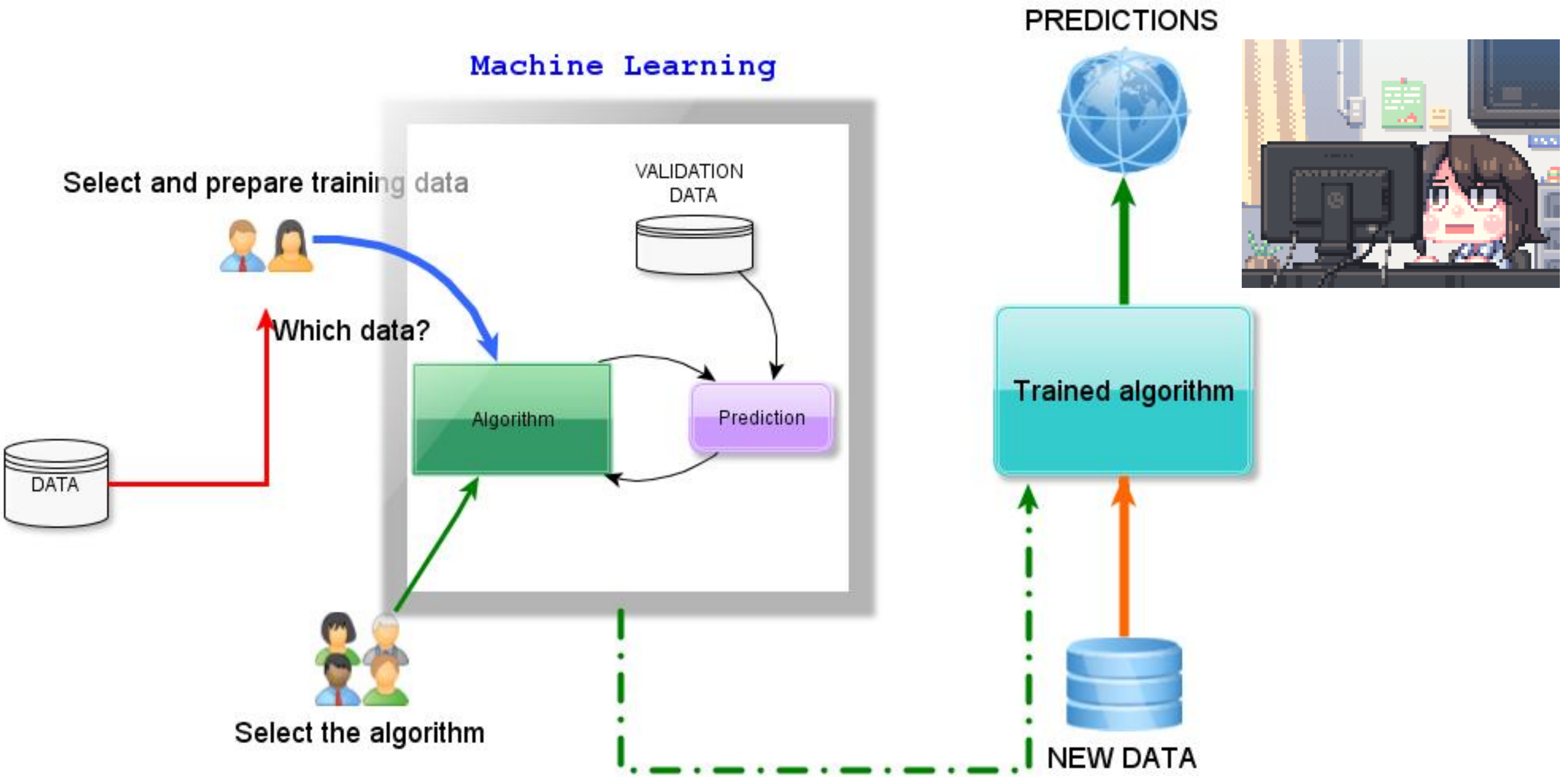
Gravity Spy, Zevin et al (2017)

Why Signal Classification?

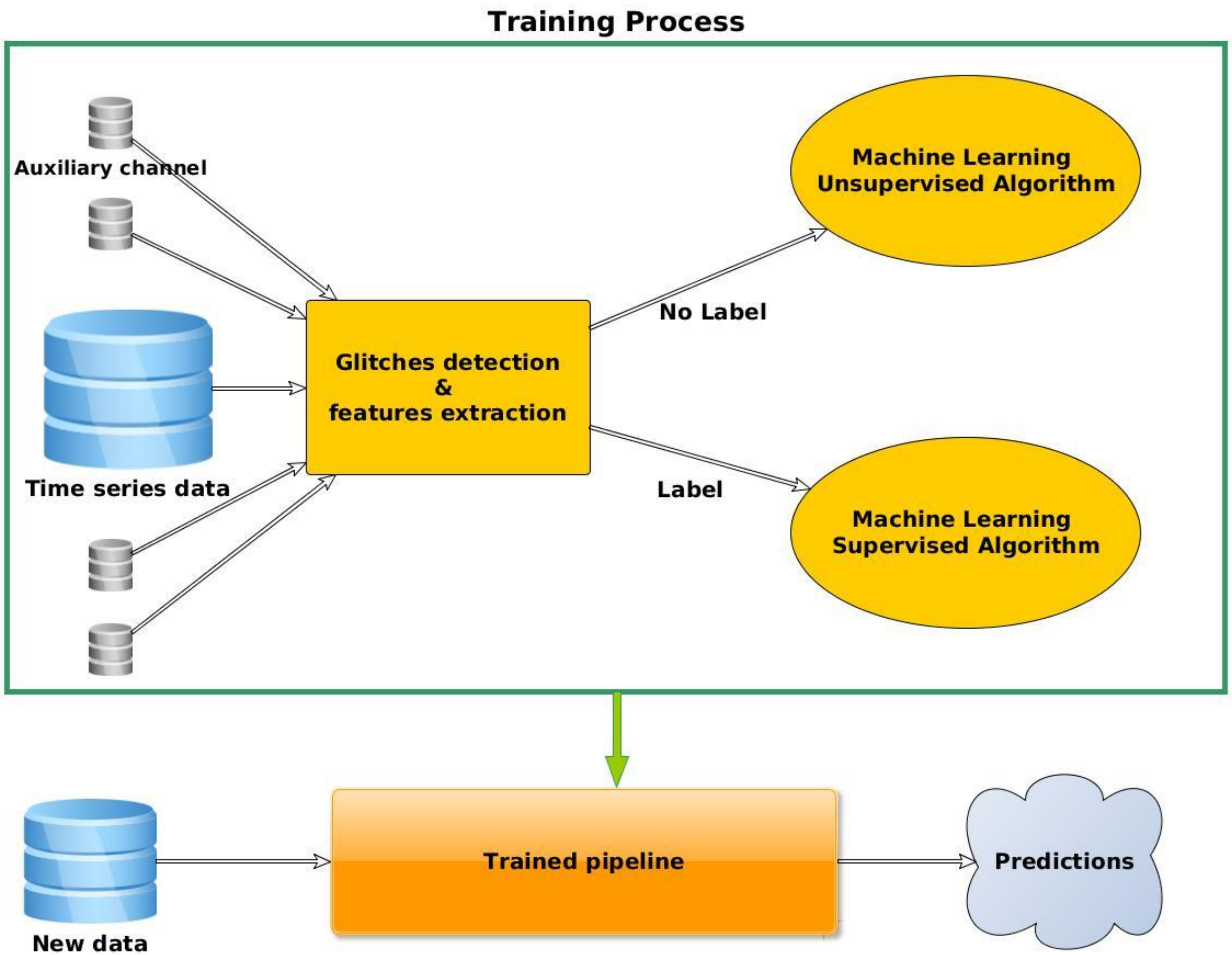
- If we are able to classify the noise events, we can clean the data in a fast and clear way
- We can help commissioners
- We can identify glitch families



Artificial Intelligence workflow



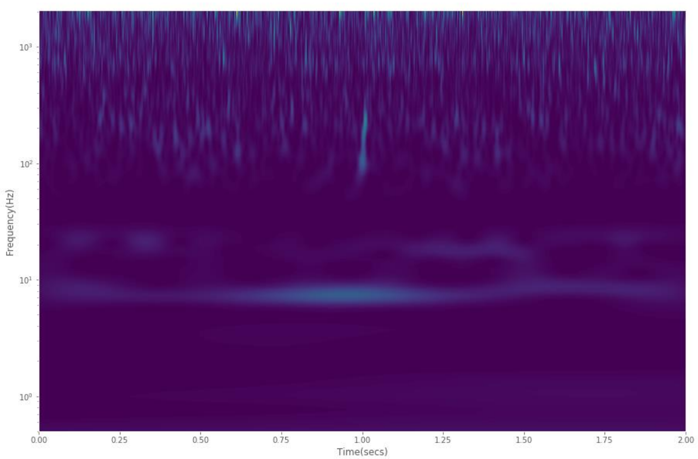
Glitch classification strategy for GW detectors



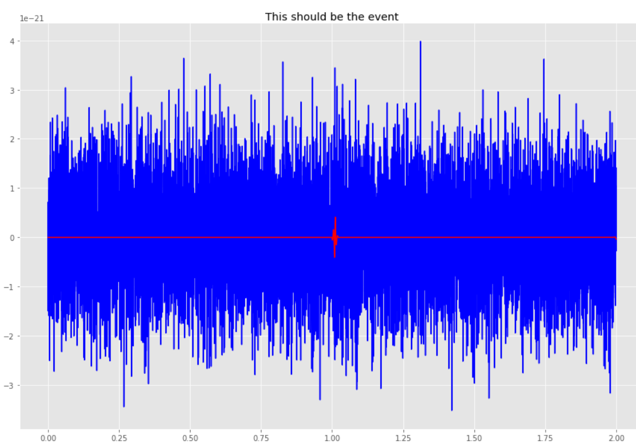
Two different approaches

- Images

- Time series



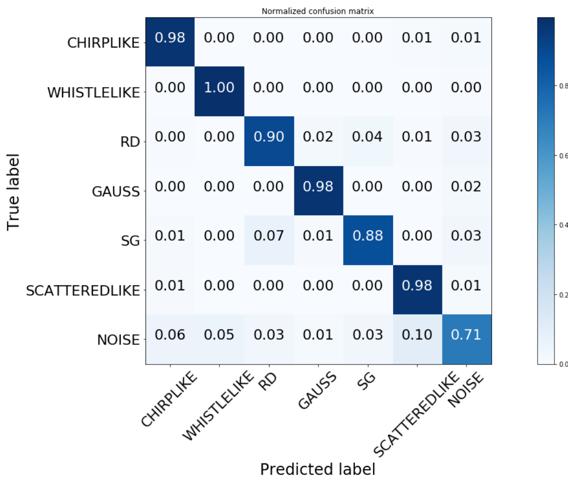
sed deep learning for classification of noise transients in gravitational wave detectors,
ano Razzano, **Elena Cuoco**, Class.Quant.Grav. 35 (2018) no.9, 095016



Wavelet-based Classification of Transient Signals for Gravitational Wave Detectors**, Elena Cuoco**, Massimiliano Razzano and Andrei Utina, #1570436751 accepted reviewed paper at EUSIPCO2018

Glitches classifcation

- Application on Simulated data
- Application on Real Data
- Time-series (Wavelet) based classification
- Image based classification with Deep Learning



Test on simulated data sets

To test the pipeline, we prepared ad-hoc simulations

Simulate colored noise using public H1 sensitivity curve

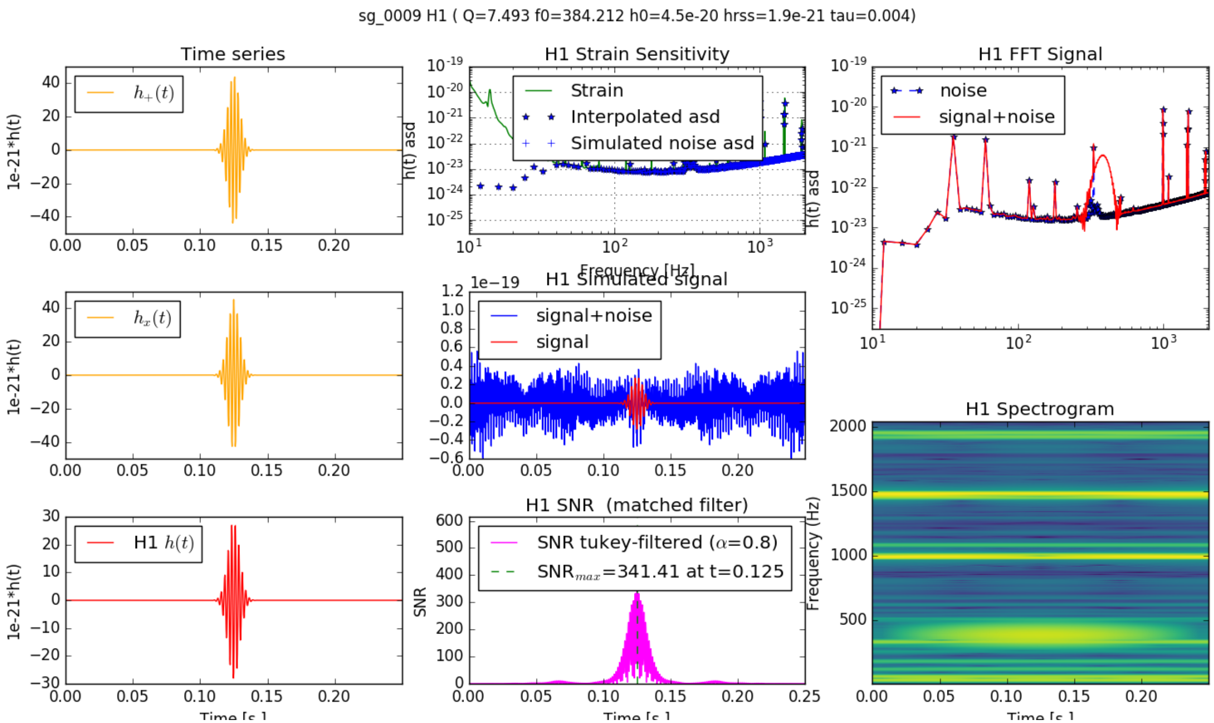
Add 6 different classes of glitch shapes

More in Filip's Tutorial

Data simulation

- Ad hoc simulations for tests (e.g. Powell+2015)
- Simulate colored noise using public sensitivity curve
- 6 classes of glitch shapes (+ NOISE one to check detection)

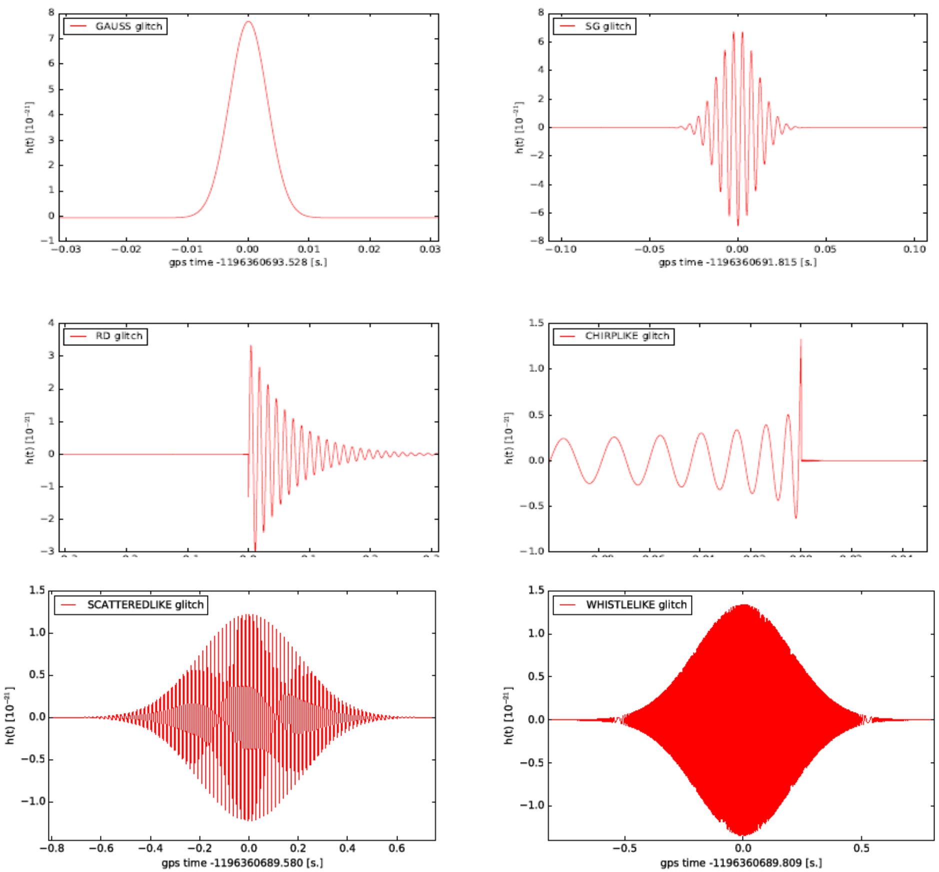
Example of H1 simulation



Razzano's courtesy

EGO EUROPEAN GRAVITATIONAL OBSERVATORY

Simulated signal families



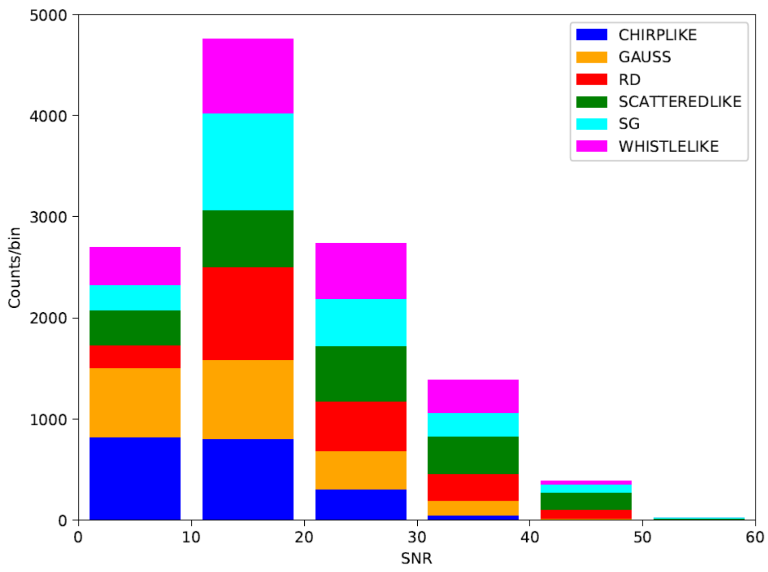
Waveform
Gaussian
Sine-Gaussian
Ring-Down
Chirp-like
Scattered-like
Whistle-like
NOISE (random)

To show the glitch time-series
here we don't show the noise contribution

Razzano M., Cuoco E. CQG-104381.R3

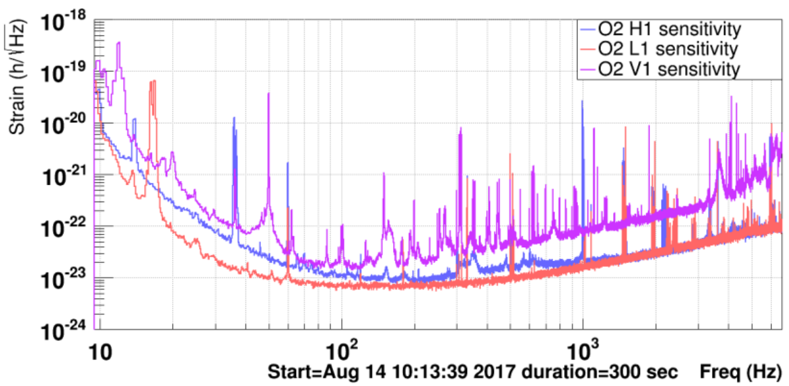
Signal distribution

- Simulated time series with 8kHz sampling rate
- Glitches distributed with Poisson statistics m=0.5 Hz
- 2000 glitches per each family
- Glitch parameters are varied randomly to achieve various shapes and Signal-To-Noise ratio

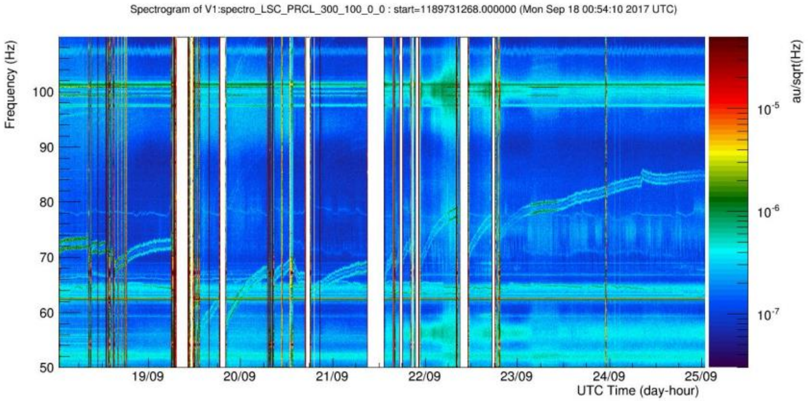


Data preprocessing

- **Many spectral features**



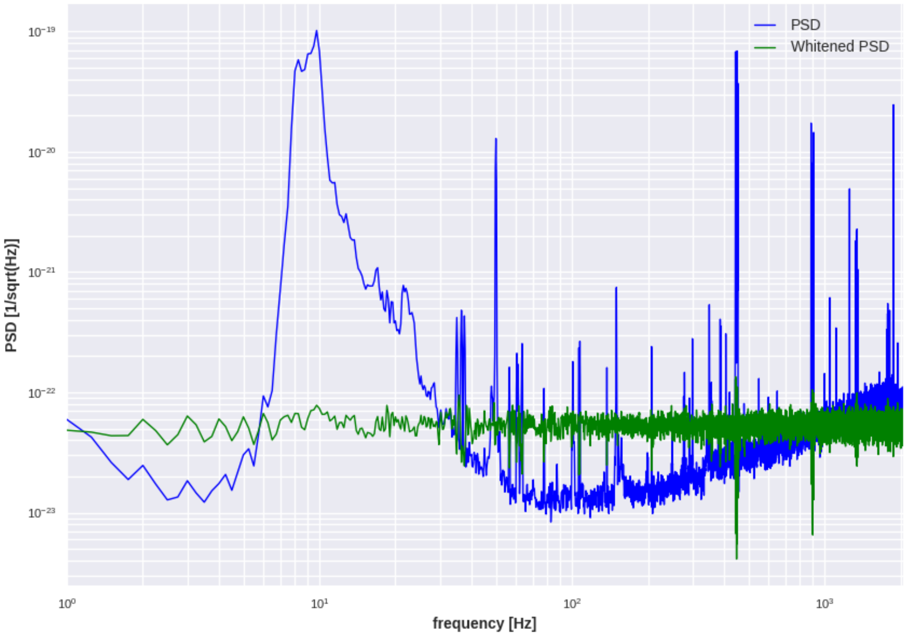
- **Non stationary and non linear noise**



Whitening in time domain

We need parametric modeling

- It can be useful for on-line application
- It can be implemented for non stationary noise
- It can catch the autocorrelation function to larger lags



AR parametric modeling

An AutoRegressive process is governed by this relation

$$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + w[n]\,,$$

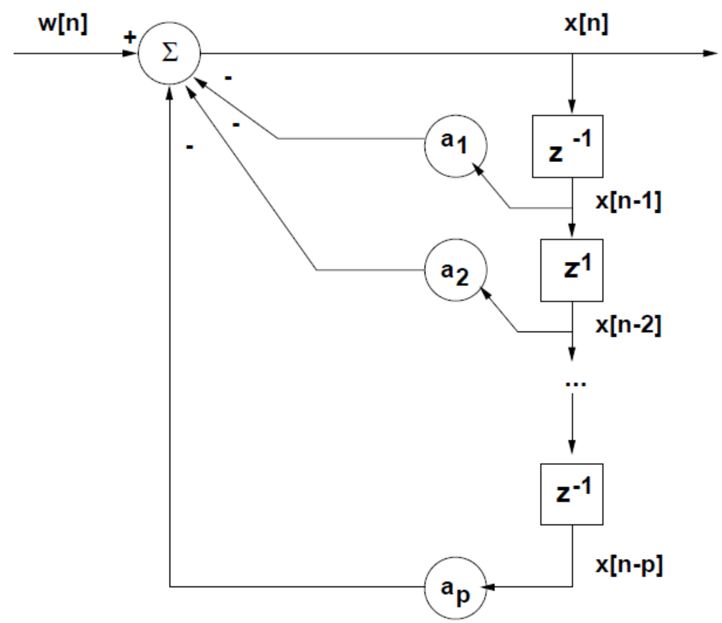
and its PSD for a process of order P is given by

$$P_{AR}(f) = \frac{\sigma^2}{|1+\sum_{k=1}^{P} a_k \exp(-i2\pi kf)|^2}$$

Kay S 1988 Modern spectral estimation: Theory and Application Prentice Hall Englewood Cliffs

Advantages of AR modeling

- Stable and causal filter: same soluti



$$\hat{x}[n] = \sum_{k=1}^{P} w_k x[n-k]\,.$$

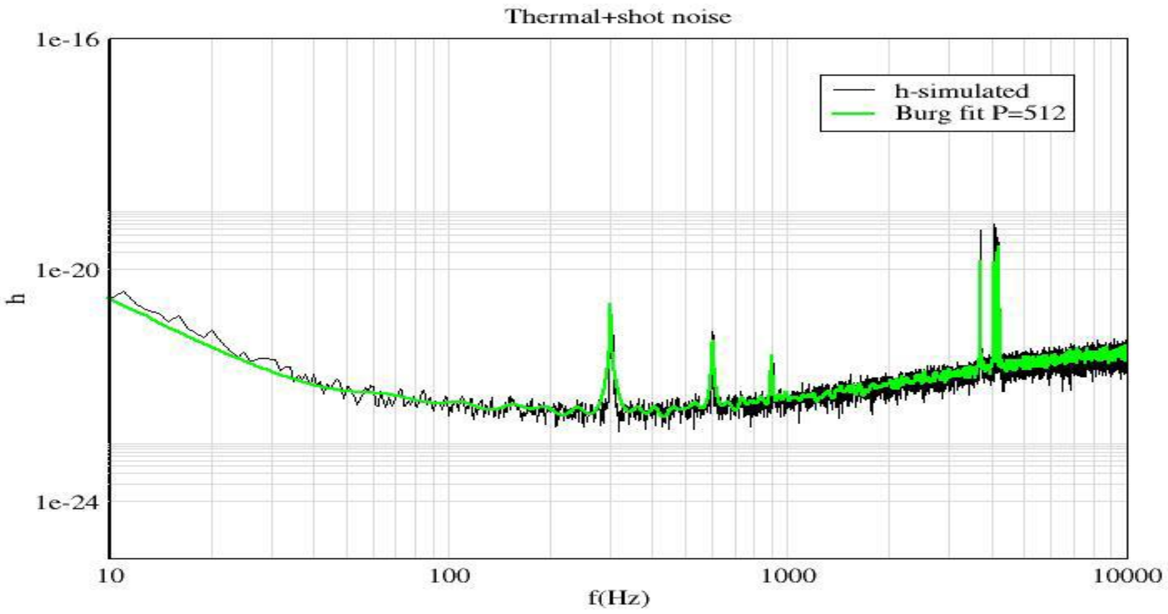
$$e[n] = x[n] - \hat{x}[n]$$

$$\varepsilon_{min} = r_{xx}[0] - \sum_{k=1}^{P} w_k r_{xx}[-k]\,,$$

$$\begin{aligned} w_k &= -a_k \\ \varepsilon_{min} &= \sigma^2 \end{aligned}$$

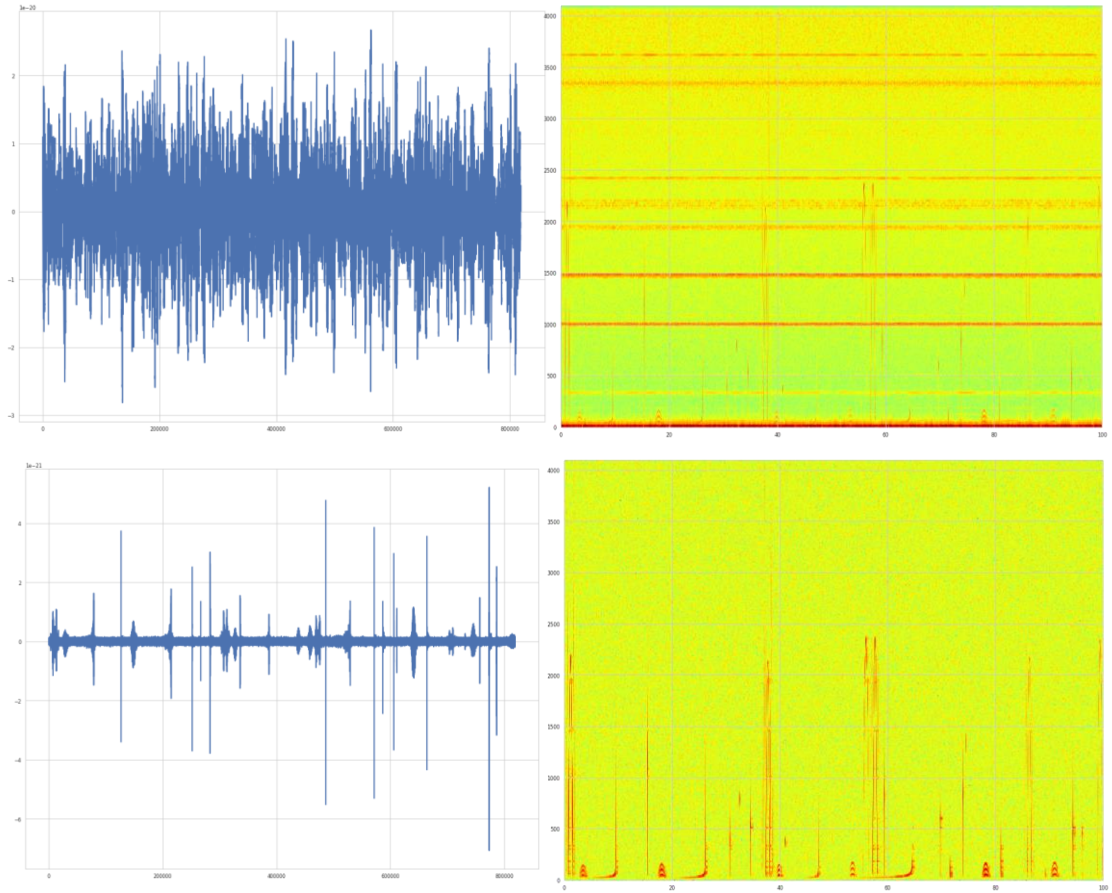
Wiener-Hopf equations

PSD AR(P) Fit



Cuoco et al. Class.Quant.Grav. 18 (2001) 1727-1752
and
Cuoco et al.Phys.Rev.D64:122002,2001

Signals in whitened data

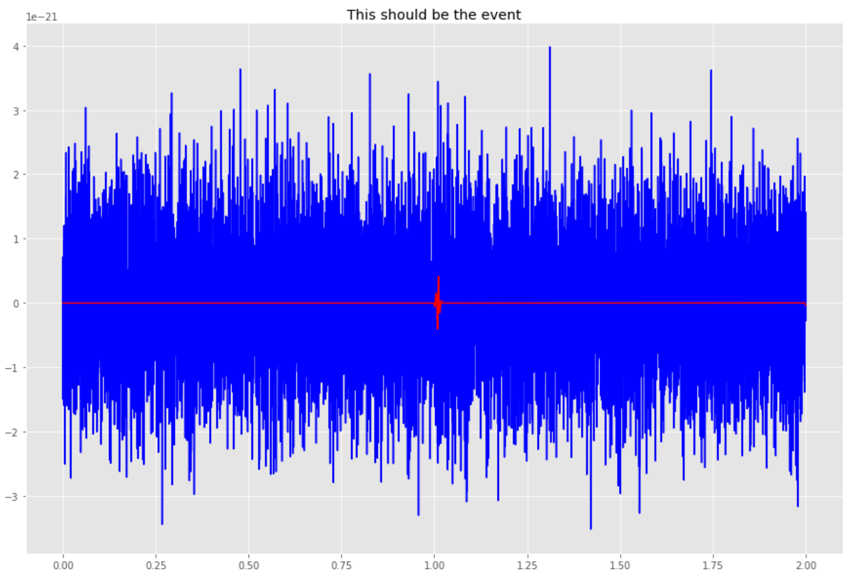


Not Whitened

Whitened

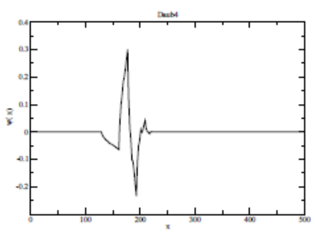
Wavelet based classification

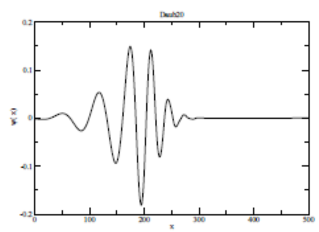
- Time series



Wavelet decomposition of time series

The wavelet transform replaces the Fourier transform sinusoidal waves by a family generated by translations and dilations of a window called a wavelet.





$$Wf(a,b) = < f, \psi_{a,b} > = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{b}} \psi^{*}\left(\frac{t-a}{b}\right) \; dt$$

Wavelet denoising

$$x_i = h_i + n_i \quad i = 0, 1, \ldots N-1$$

Wavelet transform

$$W(x) = W(h) + W(n)$$

$$t = \sqrt{2 \log N}\hat{\sigma}$$

Local noise

Threshold function

$$\hat{h} = W^{-1}(T(Wx))$$

Dohone and Johnston proposed two different thresholding strategy: the soft thresholding and the hard thresholding. Given a threshold t and w the wavelet coefficient, the hard threshold for the signal is w if $|w| > t$, and is 0 if $|w| < t$. The soft threshold for the signal is $sign(w)(|w| - t)$ if $|w| > t$ and is 0 if $|w| < t$.

Wavelet Detection filter as Event Trigger Generator

- Select highest values

$$E_s = \sqrt{\sum_{k,j} w_{k,j}^2}$$

$\propto$ Energy of the signal

- Reconstruct a proto-SNR

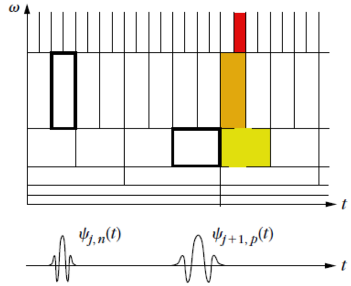
$$SNR = \frac{E_s}{\sigma}$$

$\propto$ SNR of the signal

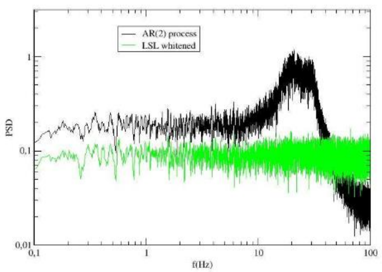
Wavelet Detection Filter (WDF) workflow

$$x_i = h_i + n_i, \quad i = 0, 1, ...N-1,$$

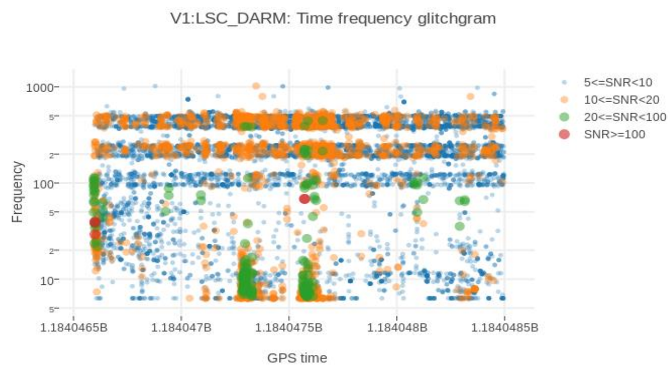
$$Wf(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{b}} \psi^* \left(\frac{t-a}{b} \right) \; dt.$$



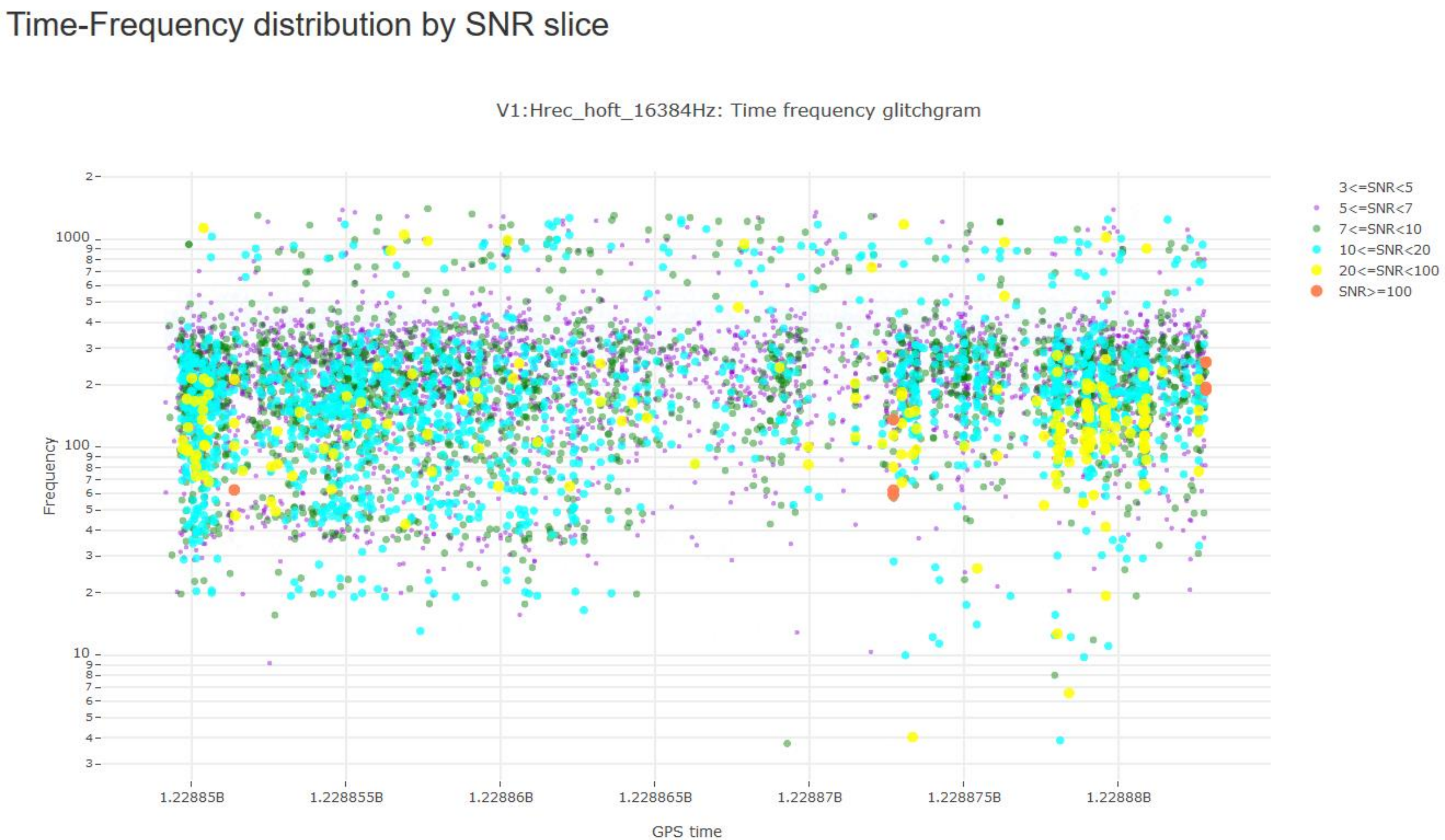
Data → Whitening in time domain → Wavelet transform → De-noising → Parameter estimation → Trigger list



$$\hat{h}_i = W^{-1}(t[W(x_i)]).$$

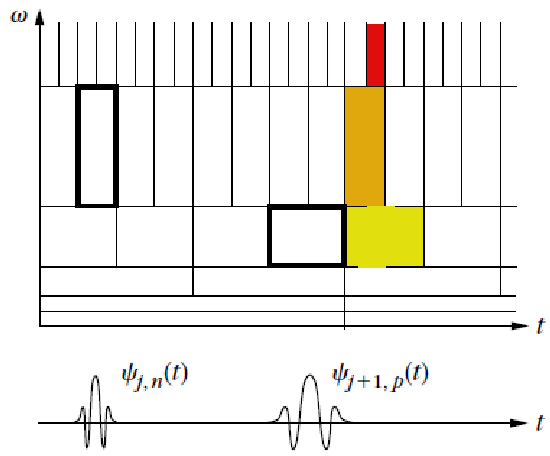


Glitch-gram


Time-Frequency distribution by SNR slice
V1:Hrec_hoft_16384Hz: Time frequency glitchgram
Frequency
GPS time
1.22885B
1.228855B
1.22886B
1.228865B
1.22887B
1.228875B
1.22888B
3<=SNR<5
5<=SNR<7
7<=SNR<10
10<=SNR<20
20<=SNR<100
SNR>=100

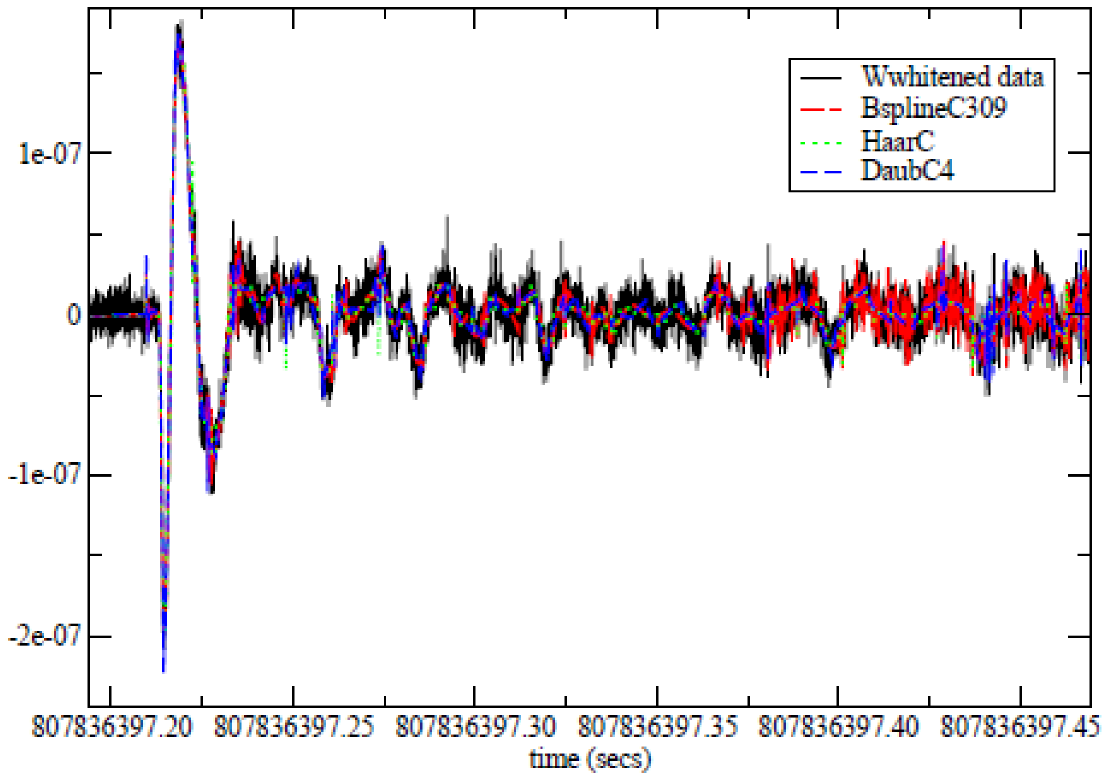
WDF waveform extraction

- Wavelet transform in the selected window size
- Retain only coefficients above a fixed threshod (Donoho-Johnston denoise method)
- Create a metrics for the energy using the selected coefficients and give back the trigger with all the wavelet coefficients.
- In the wavelet plane, select the highest values to build the event
- Inverse wavelet transform
- Estimate mean and max frequency and snr max of the cleaned event

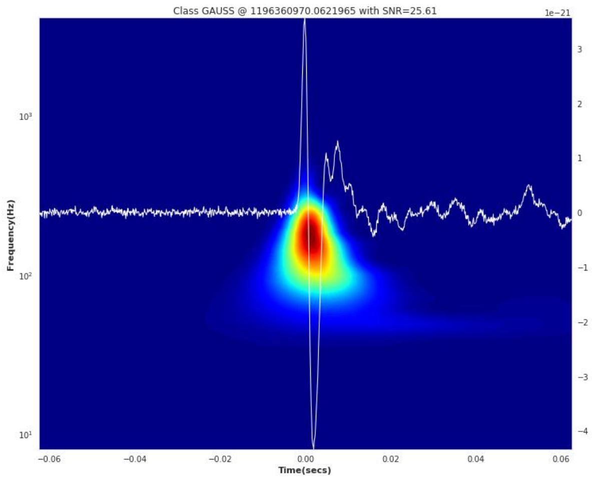


Gps, duration, snr, snr@max, freq_mean, freq@max, wavelet type triggered + corresponding wavelets coefficients.

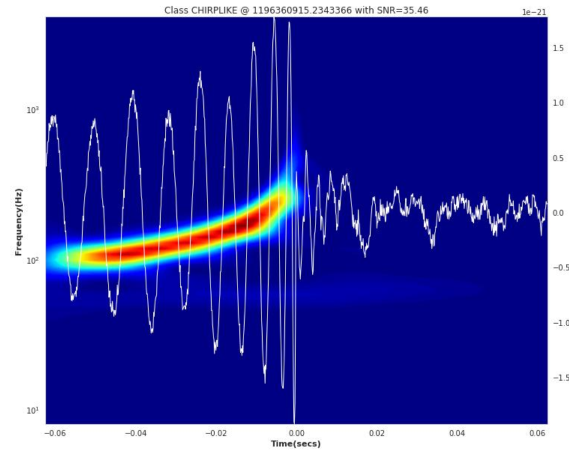
Waveform reconstruction

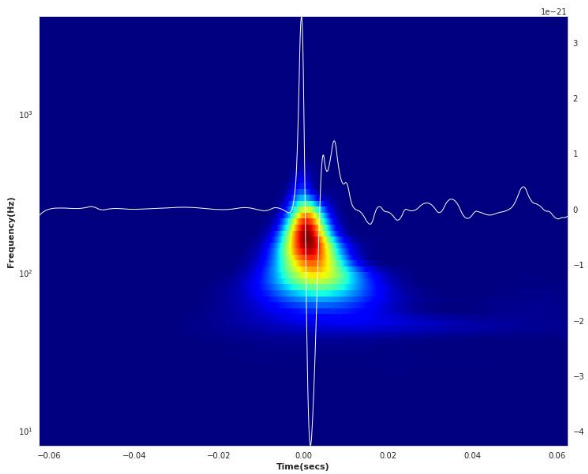


Waveform reconstruction: example

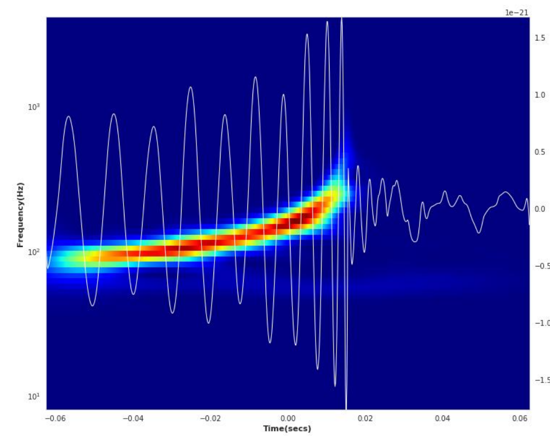


Injected



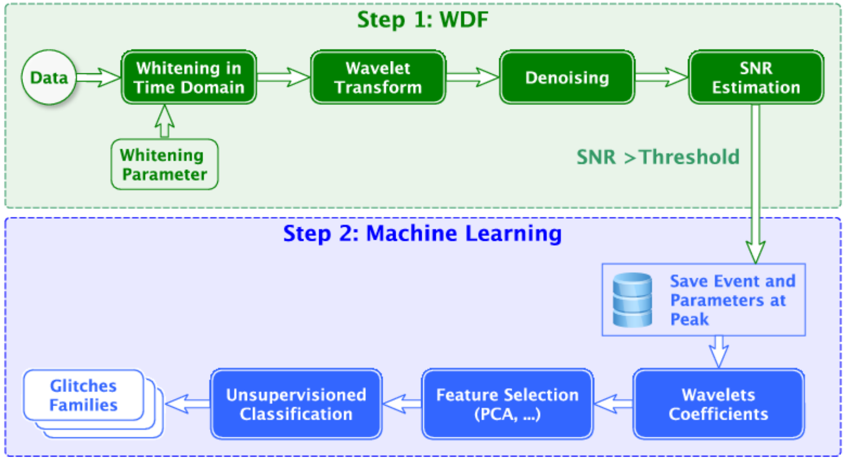


Detected

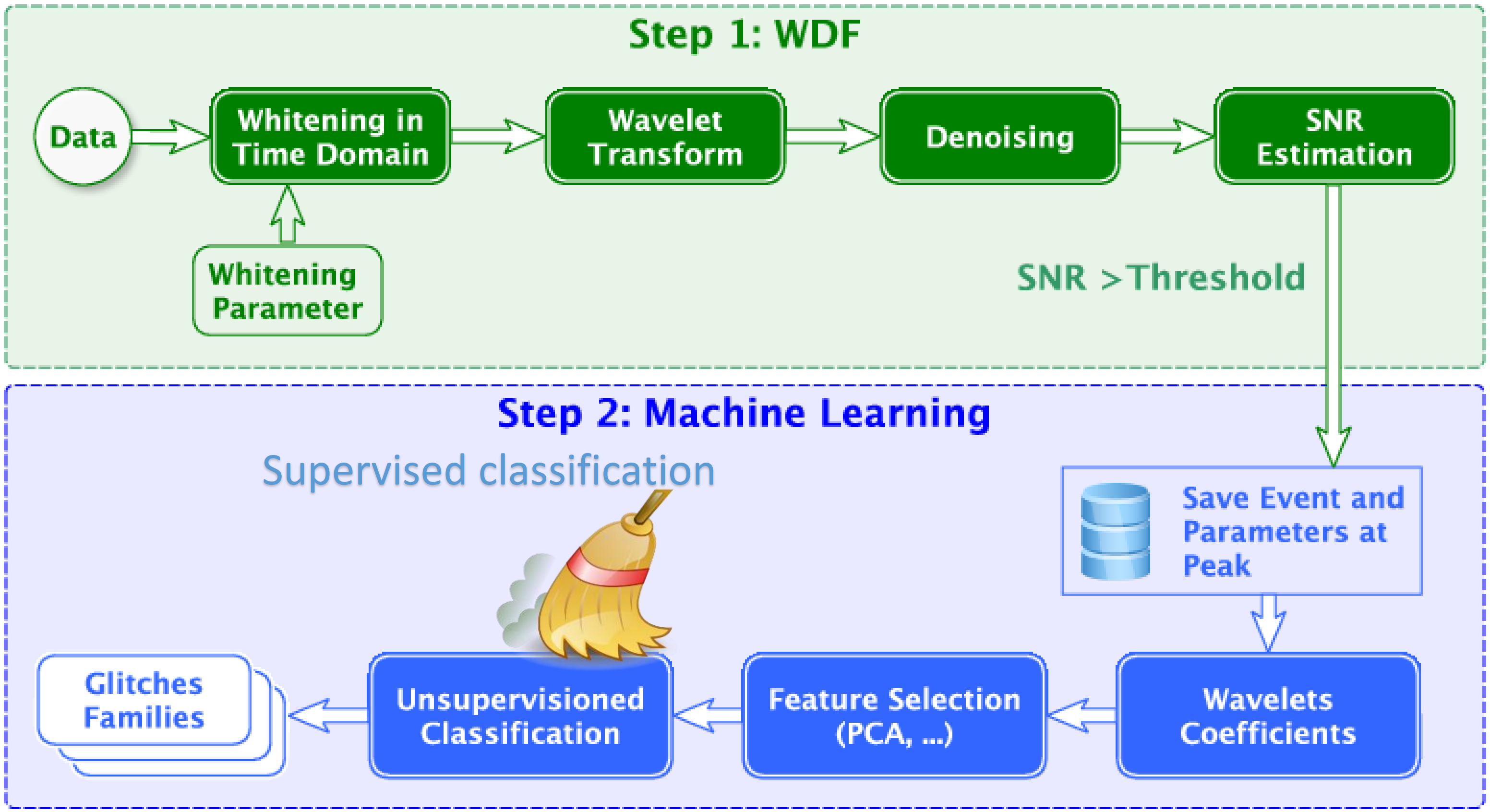


Glitch classification, past works

- Unsupervised on Simulated data:
  - Classification methods for noise transients in advanced gravitational-wave detectors
    Jade Powell, Daniele Trifirò, **Elena Cuoco**, Ik Siong Heng, Marco Cavaglià, Class.Quant.Grav. 32 (2015) no.21, 215012
- Unsupervised on Real data (ER7):
  - Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data, Jade Powell, Alejandro Torres-Forné, Ryan Lynch, Daniele Trifirò, **Elena Cuoco**, Marco Cavaglià, Ik Siong Heng, José A. Font, Class.Quant.Grav. 34 (2017) no.3, 034002

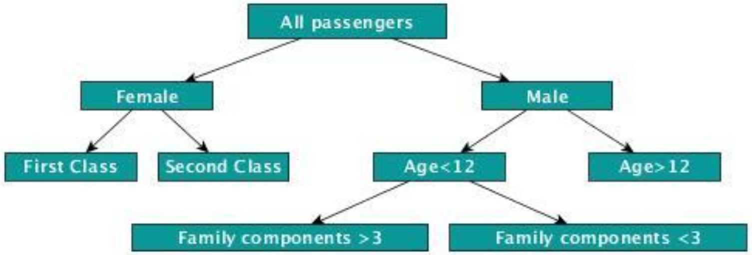


Wavelet Detection Filter and XGBoost (WDFX)



Supervised Classification: eXtreme Gradient Boosting

- https://github.com/dmlc/xgboost
- Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016
- XGBoost originates from research project at University of Washington, see also the Project Page at UW.



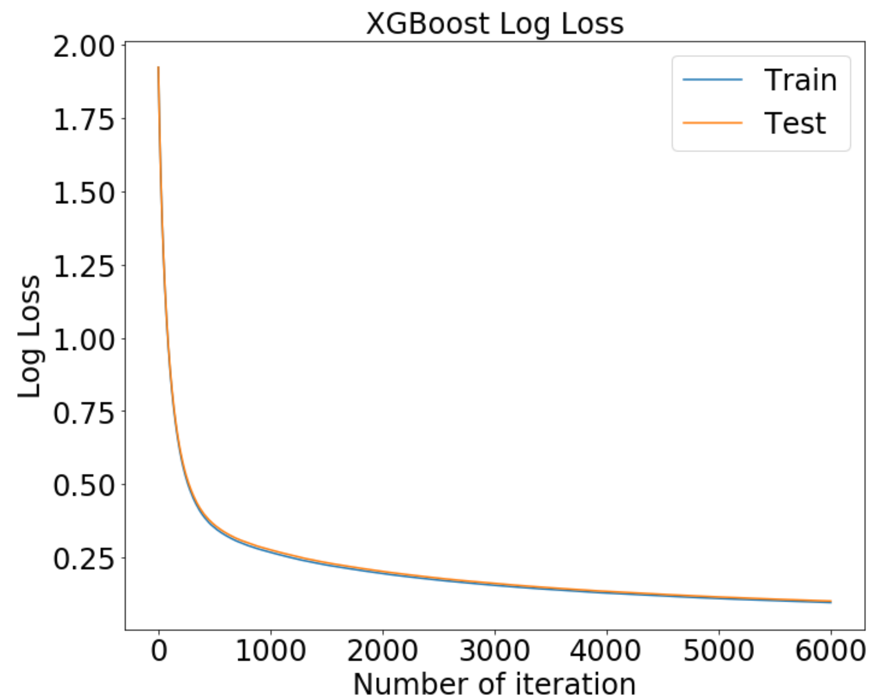
Tree Ensemble

$$y_n = \sum_{k=1}^{K} f_k(x_n)$$

dmlc XGBoost

Xgboost

$$L = -\frac{1}{N}\sum_{1}^{N}((y_i \log(p_i) + (1-y_i)(log(1-p_i)) + \Omega$$



Train/validation/test set: 70/15/15

task	Classes	Learning-rate	Max_depth	estimators
Binary	2	0.01	7	5000
Multi-label	7	0.01	10	6000

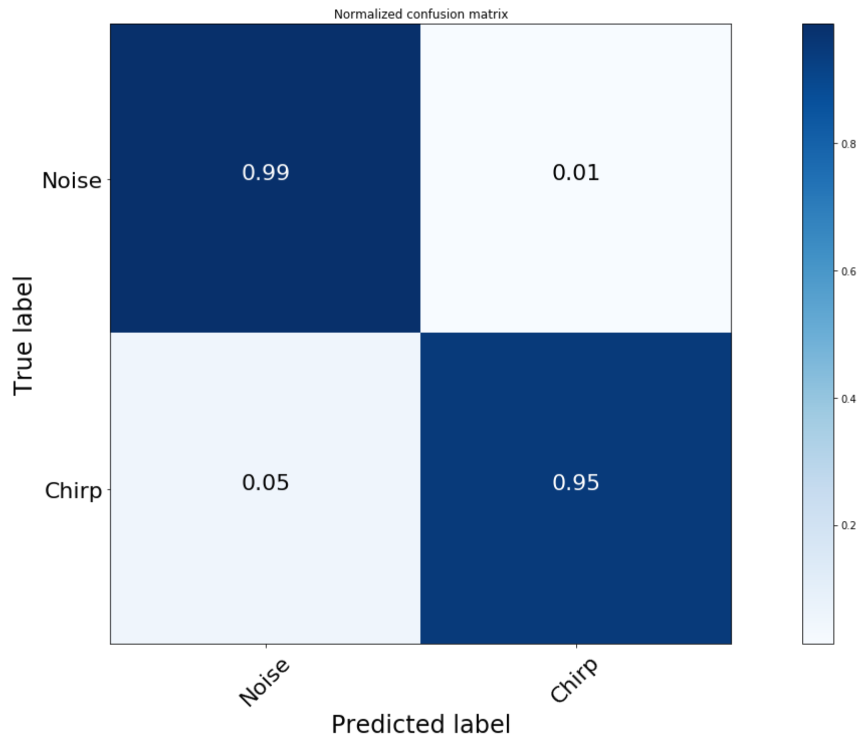
WDFX: Binary Classification Results

Chirp-like signals

OR

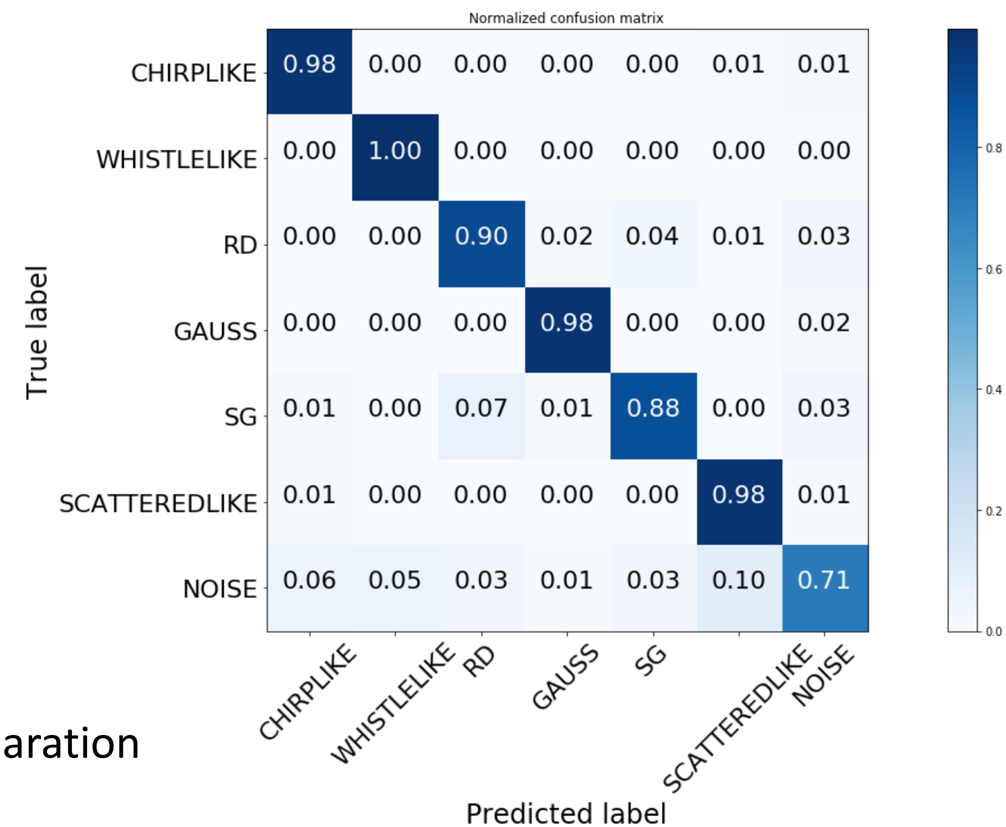
Noise

Overall accuracy >98%



WDFX Results: Multi-Label Classification

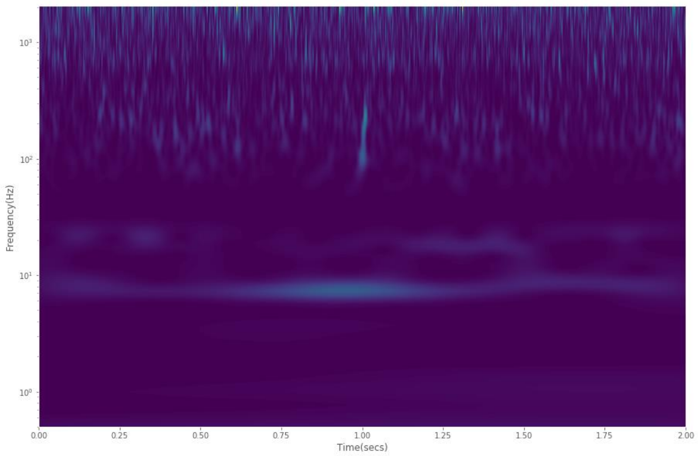
Overall accuracy >93%



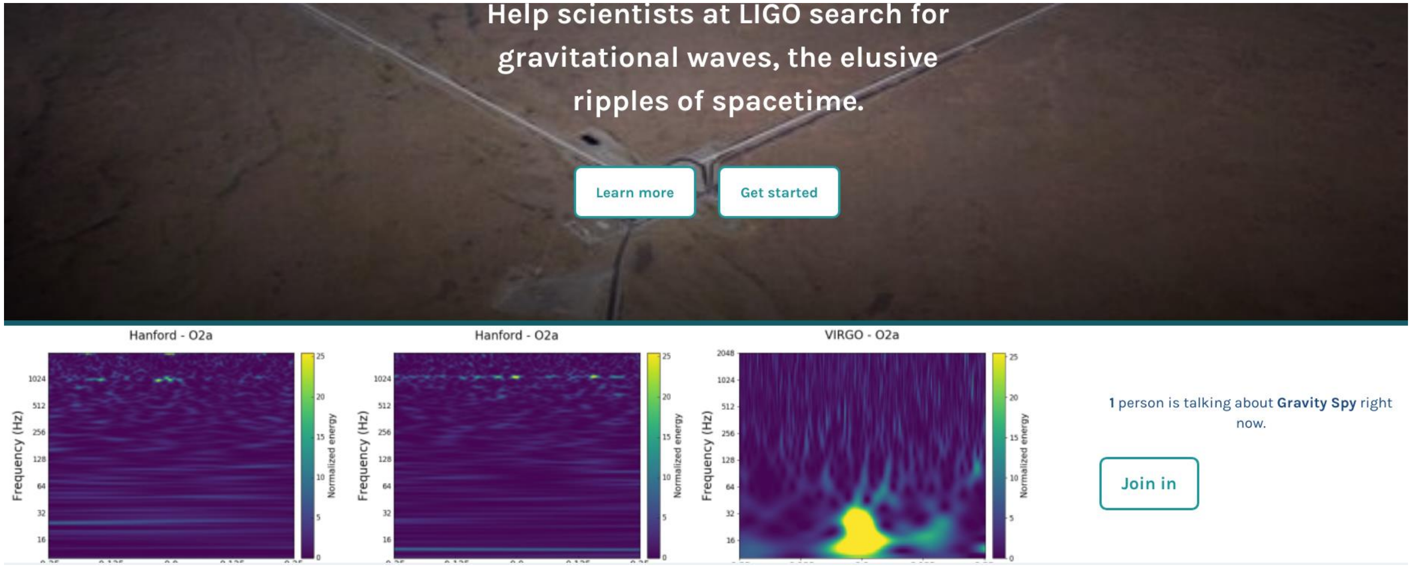
Cuoco, Morawski, Razzano in preparation

Image-based classification

- Images



Glitch & Citizen science: GravitySpy

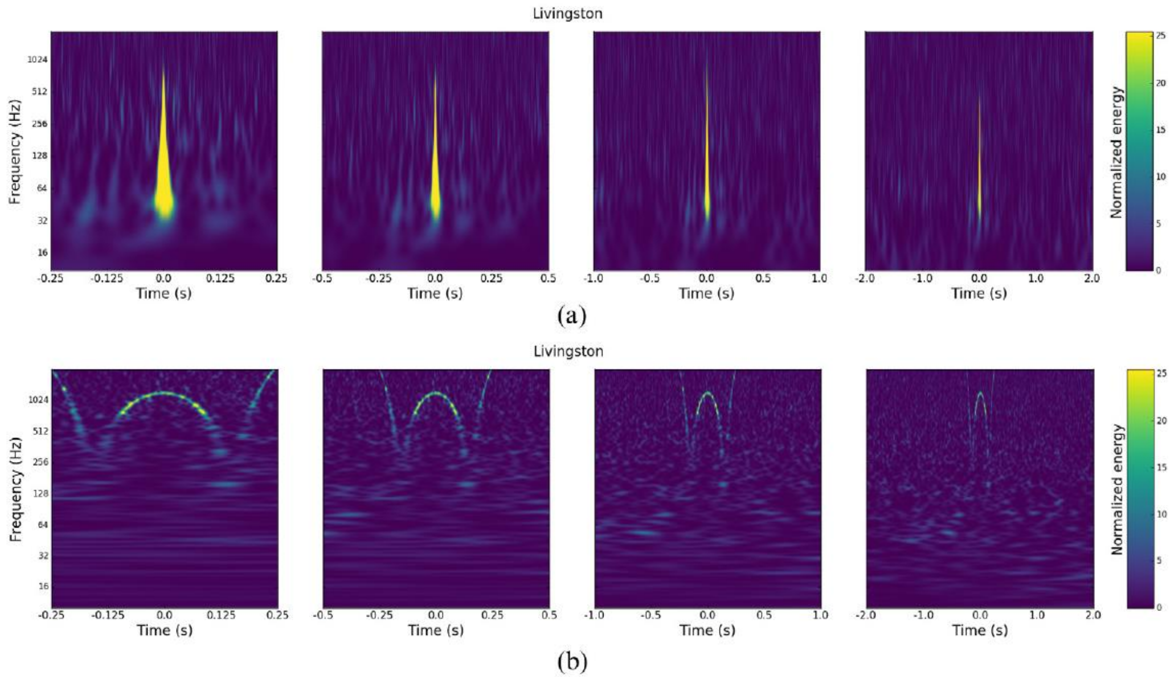


[Www.gravityspy.org](http://www.gravityspy.org)

Citizen scientists contribute to classify glitches

More details in Zevin+17

Sample glitch gallery

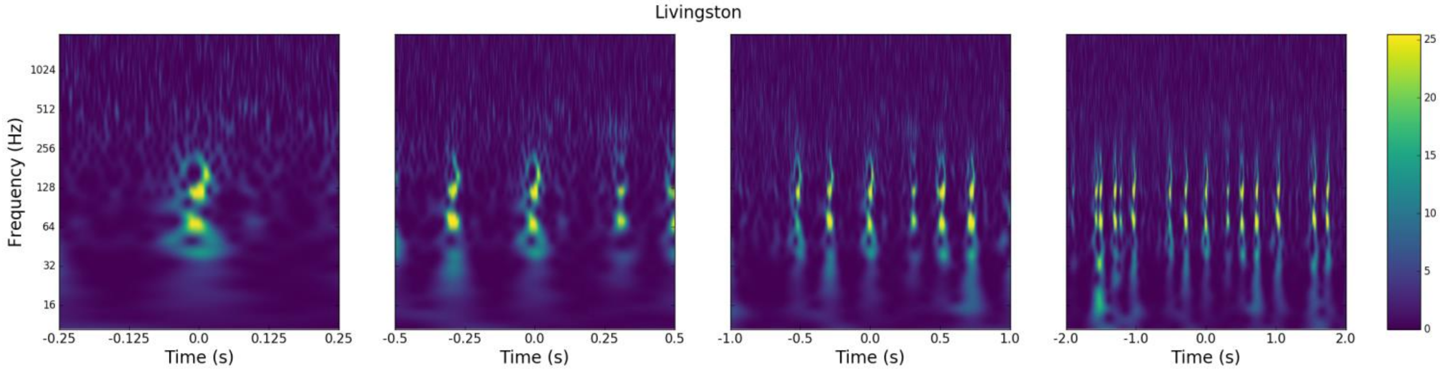


Blip glitches

Whistle glitches

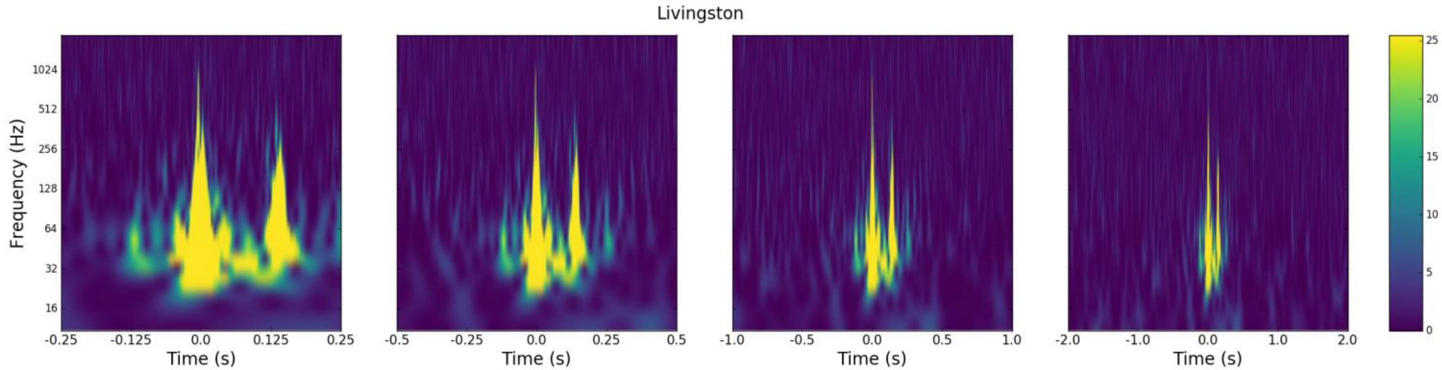
Examples of time-frequency glitch morphology (Zevin+17)

Sample glitch gallery



Helix glitches

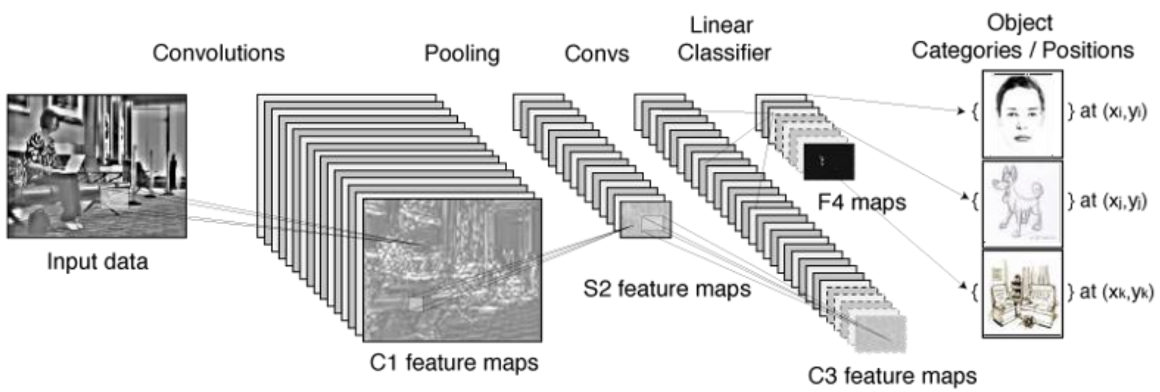
Koi fish glitches



EGO EUROPEAN GRAVITATIONAL OBSERVATORY

Deep learning for Glitch Classification

- Many approaches to data: we choose image classification of time frequency images
- The architecture is based on Convolutional deep Neural Networks (CNNs).
- CNNs are more complex than simple NNs but are optimized to catch features in images, so they are the best choice for image classification



Pipeline structure

Input GW data

- Image processing
- Time series whitening
- Image creation from time series (FFT spectrograms)
- Image equalization & contrast enhancement

Classification

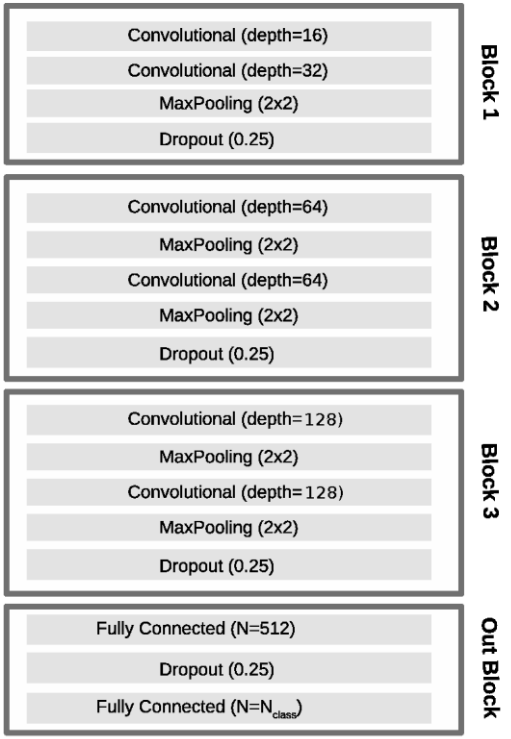
- A probability for each class, take the max
- Add a NOISE class to crosscheck glitch detection

Network layout

- Tested various networks, including a 4-block layers

Run on GPU Nvidia GeForce GTX 780

- 2.8k cores, 3 Gb RAM)
- Developed in Python + CUDA-optimized libraries



Building the images

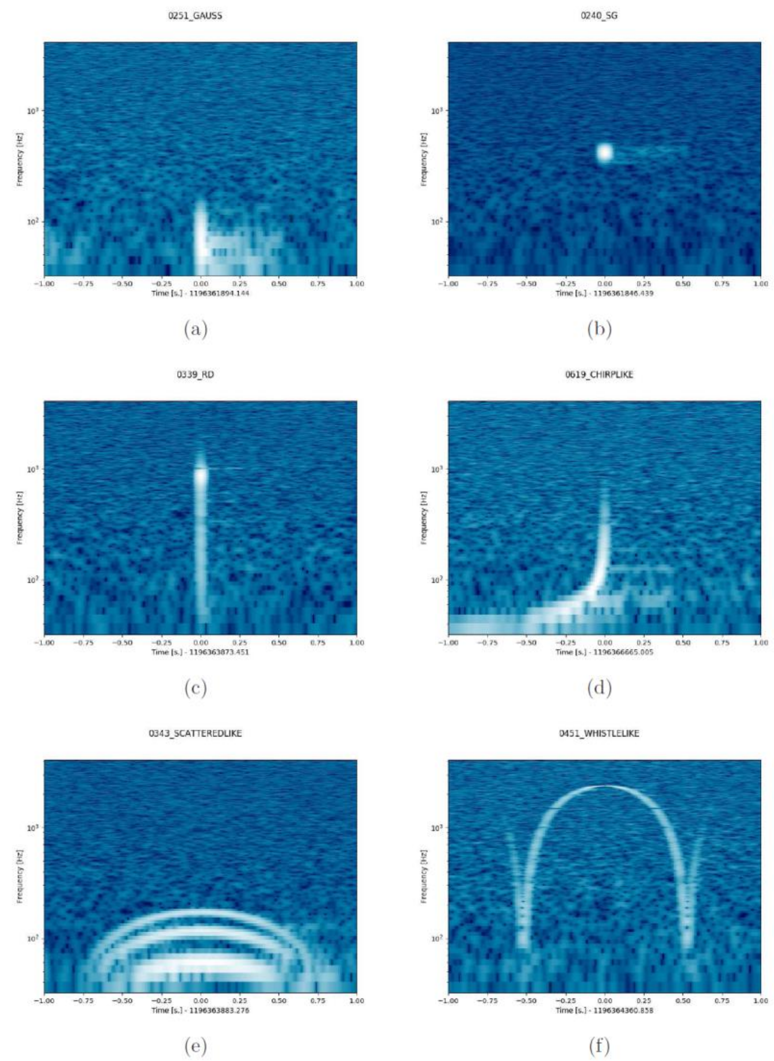
Spectrogram for each image

2-seconds time window to highlight fatures in long glitches

Data is whitened

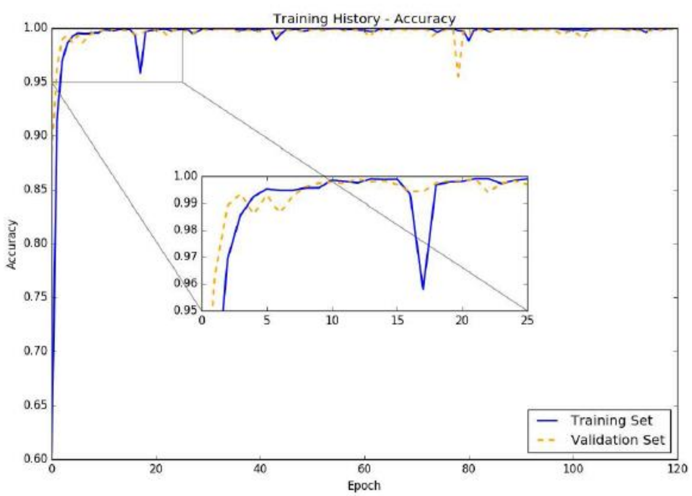
Optional contrast stretch

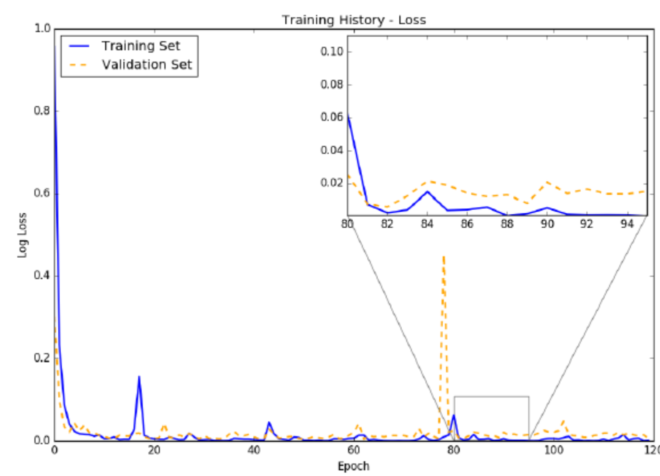
Simulations now available
on FigShare



Training the CNN

- Datasets of 14000 images
- Training/validation/test → 70/15/15
- Image size 241px x 513px
- Reduced the images by a factor 0.55 due to memory constraints
- Use validation set to tune hyperparameters
- On our hardware, training time ~8 hrs for ~100 epochs
- When training is done, classification requires ~1 ms/image (on our configuration)





Classification Results

We compared classification performances with simpler architectures

Metric	Accuracy	Precision	Recall	F1 score	Log loss
SVM	0.971	0.972	0.971	0.971	0.08
Shallow CNN	0.986	0.986	0.986	0.986	0.04
1 CNN block	0.991	0.991	0.991	0.991	0.02
3 CNN blocks	0.998	0.998	0.998	0.998	0.008

Linear Support Vector Machine

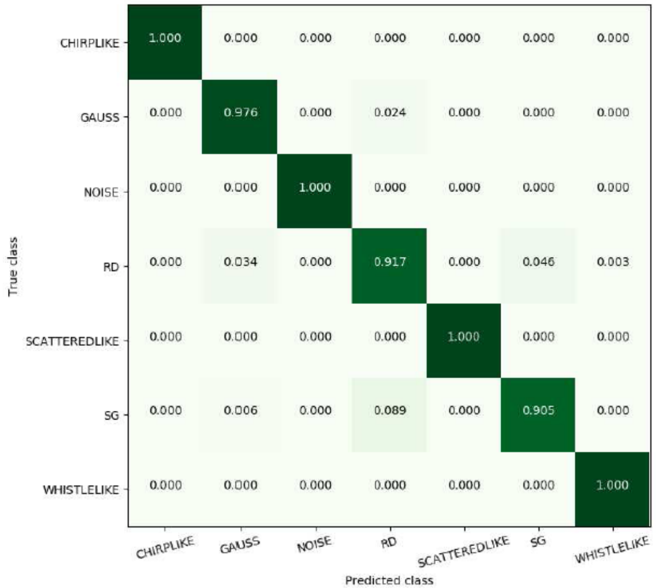
CNN with 1 hidden layer

CNN with one block
(2 CNNs+Pooling&Dropout)

Deep 4-blocks CNNs

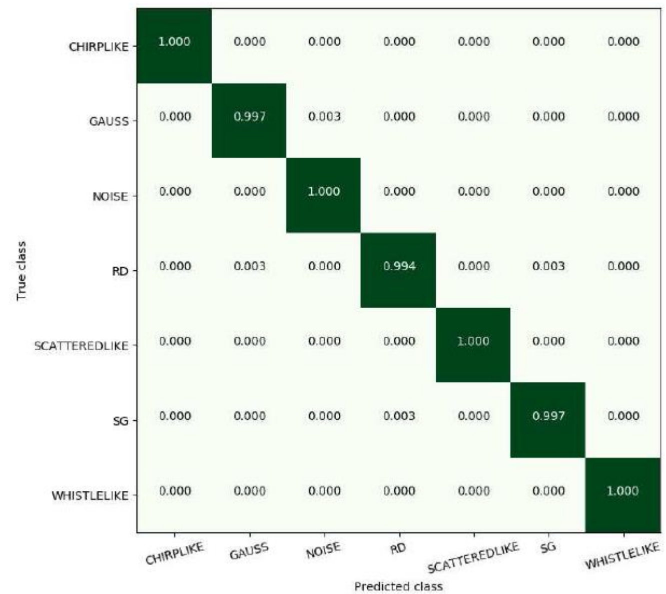
Classification accuracy

Normalized Confusion Matrix



SVM

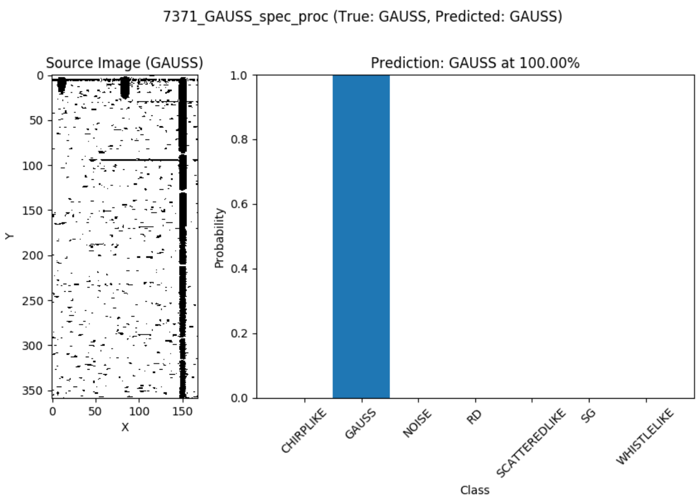
Deep CNN

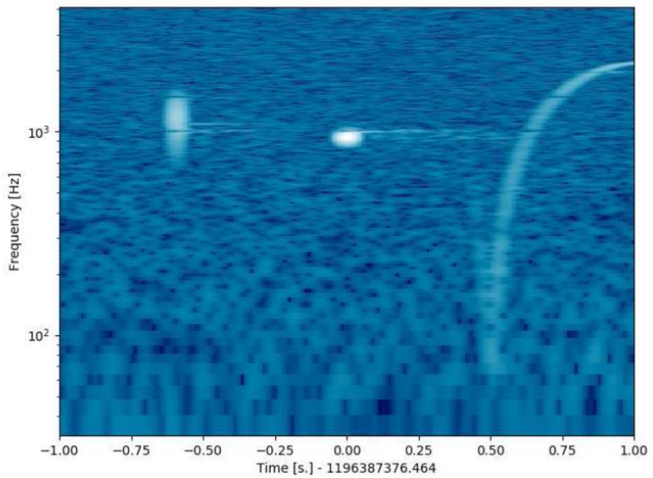


Deep CNN better at distinguishing similar morphologies

Example of classification results

Some cases of more glitches in the time window, always identify the right class





100% Sine-Gaussian

More details in
Razzano & Cuoco 2018, CQG,35,9

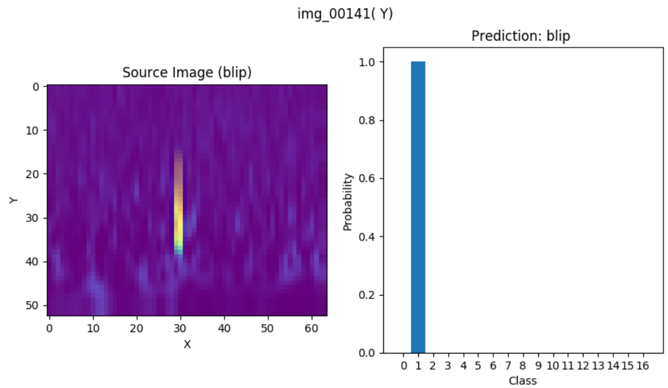
Real data: O1 run

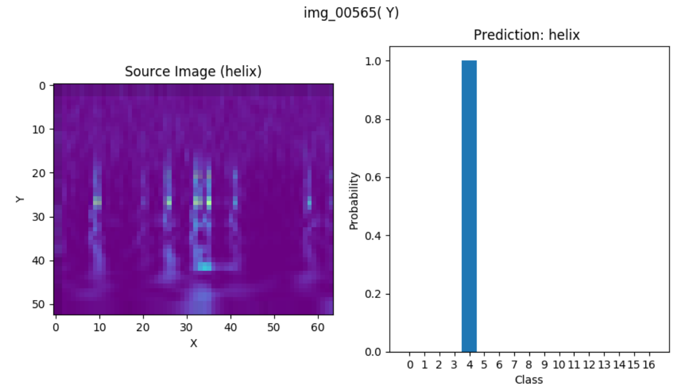
Dataset from GravitySpy images

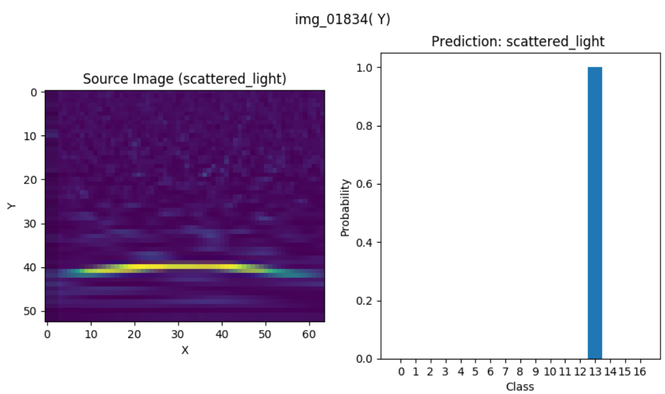
Glitch name	# in H1	# in L1
Air compressor	55	3
Blip	1495	374
Chirp	34	32
Extremely Loud	266	188
Helix	3	276
Koi fish	580	250
Light Modulation	568	5
Low_frequency_burst	184	473
Low_frequency_lines	82	371
No_Glitch	117	64
None_of_the_above	57	31
Paired doves	27	-
Power_line	274	179
Repeating blips	249	36
Scattered_light	393	66
Scratchy	95	259
Tomte	70	46
Violin_mode	179	-
Wandering_line	44	-
Whistle	2	303

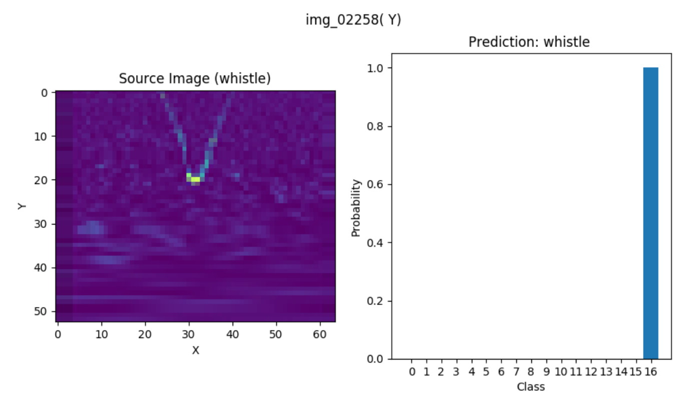
EGO EUROPEAN GRAVITATIONAL OBSERVATORY

Examples of classification









Results

Confusion Matrix (Normalized)

True class

	1080lines	1400ripples	air_compressor	blip	chirp	extremely_loud	helix	koi_fish	light_modulation	low_frequency_burst	low_frequency_lines	no_glitch	none_of_the_above	paired_doves	power_line	repeating_blips	scattered_light	scratchy	tomte	violin_mode	wandering_line	whistle
1080lines	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1400ripples	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
air_compressor	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
blip	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
chirp	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
extremely_loud	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
helix	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
koi_fish	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
light_modulation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
low_frequency_burst	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
low_frequency_lines	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.88	0.05	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00
no_glitch	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
none_of_the_above	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.69	0.00	0.00	0.00	0.00	0.08	0.15	0.00	0.00	0.08
paired_doves	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
power_line	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
repeating_blips	0.00	0.00	0.00	0.05	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00
scattered_light	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.01	0.00	0.00	0.00	0.00
scratchy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
tomte	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
violin_mode	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00
wandering_line	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
whistle	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.91

Full CNN stack

Consistent with Zevin+2017

EGO EUROPEAN GRAVITATIONAL OBSERVATORY

WaveFier

A prototype for real time Gravitational Wave transient signal classifier

Elena Cuoco (**EGO**) *Scientific Supervisor*

Emanuel Marzini, Filip Morawski,

Alessandro Petrocelli, Alessandro Staniscia, Silvana Muscella (**Trust-IT**)

Trust-IT Services

Communicating ICT to markets

H2020-ASTERICS project brings together for the first time scientists and communities from astronomy, astrophysics, particle astrophysics & big data.

http://www.asterics2020.eu

H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).

H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).

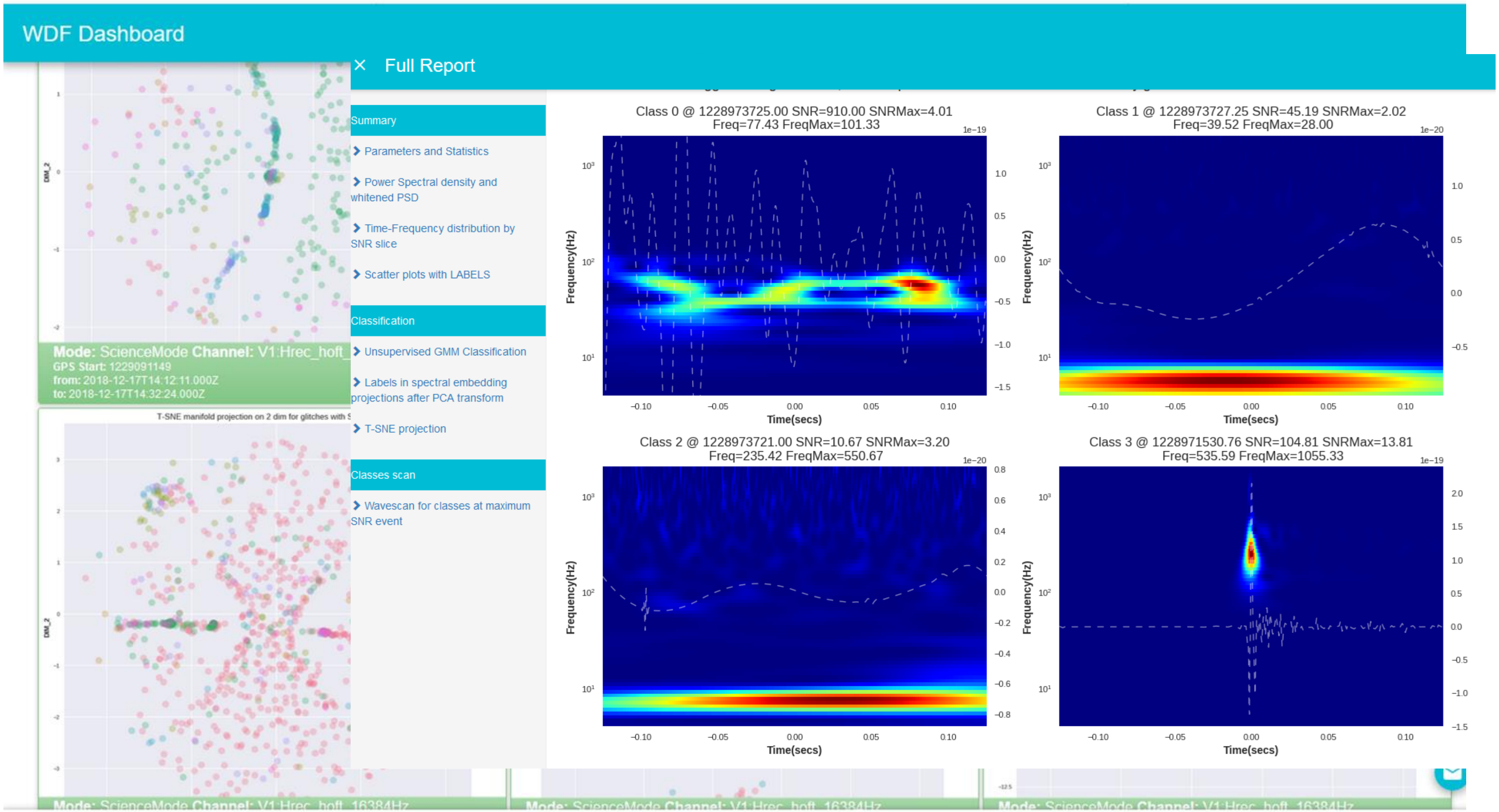
Why Wavefier

- It would be extremely useful to have an online pipeline for automatic identification and classification of transient signals for Gravitational Wave detectors and their direct database inclusion**.**
- We wanted to setup a protype for a framework where inserting ML pipeline, using new technologies, platform indipendent
- We want to have a system platform indipendent. Made test on cloud system

Wavefier: Key Objectives

- Setup a prototype for a **real time** pipeline for the detection of transient signals and their **automatic** classification
- Best practice for **software management**
- Test different software architecture solutions to prototype a **scalable** pipeline for **big data** analysis in GW context.
- **Interoperability** and access to data and services
- **ICT services** supporting research infrastructures
- Use of **data in network** infrastructures and services
- Big data and **Machine Learning**
- Test on **cluster**

What already exists (https://wdf.virgo-gw.eu/)



Machine Learning pipelines

- We worked on an easy solution: using the **features extracted by WDF** pipeline such as meta parameters (freq, SNR, duration) and wavelet coefficients or reconstructed waveforms
- We developed Machine Learning pipeline based on two different algorithms realizing two types of data analysis:
  - **Classification**
  - **Clusterization**

Both algorithms were trained on artificial data, reach in various glitches and GW signals.

- Trained the moment, the system is deployed.
- The initial architectures have been chosen after several tests as the one reaching the highest performance.
- However the implementation allows for further development via modification of configuration files.

Machine Learning classification

Classification is realized through

1 Dimensional Convolutional Neural Networks (**CNN1d**)

As an input data, the algorithm uses **reconstructed waveforms** generated by WDF.

The output is one of 7 labels:

- 6 types of glitches
- GW signal (so called “chirp”)

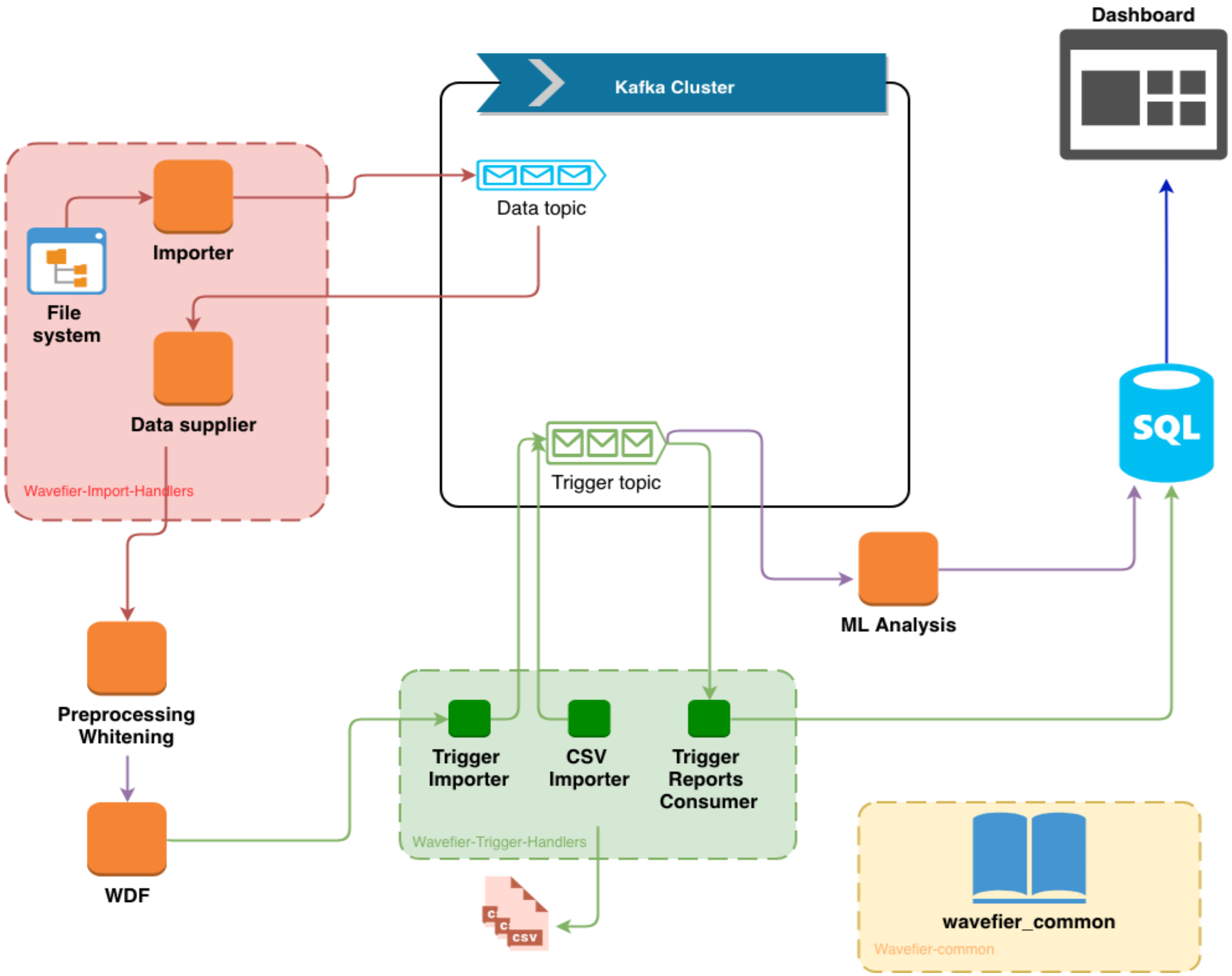
Machine Learning clusterization

Clusterization is realized through **Autoencoders** based on Artificial Neural Networks.

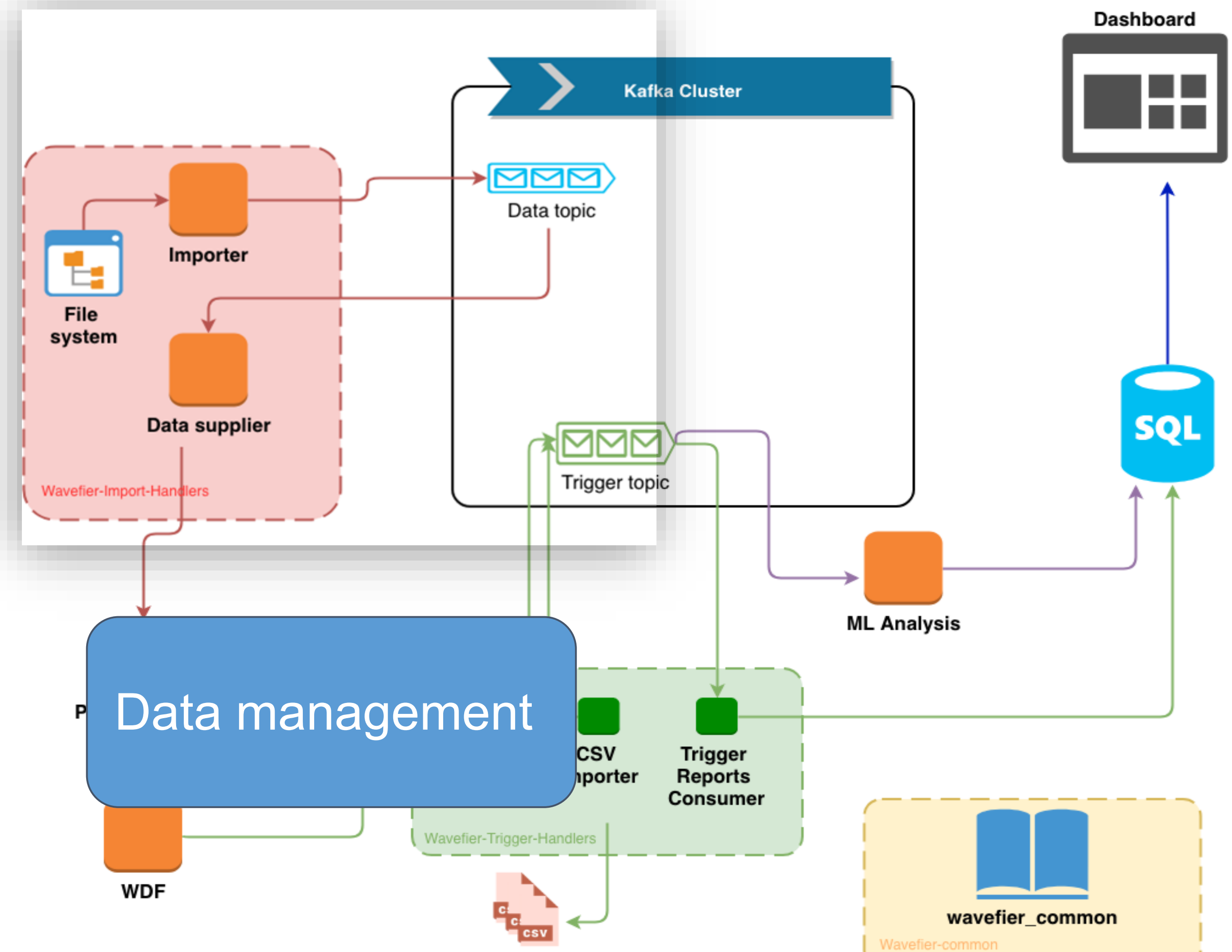
- The algorithm processes the reconstructed waveforms generated by WDF trying to find their hidden (latent) representation.
- The output is a set of parameters describing each signal in latent space (which might help in unsupervised classification).

Architecture overview

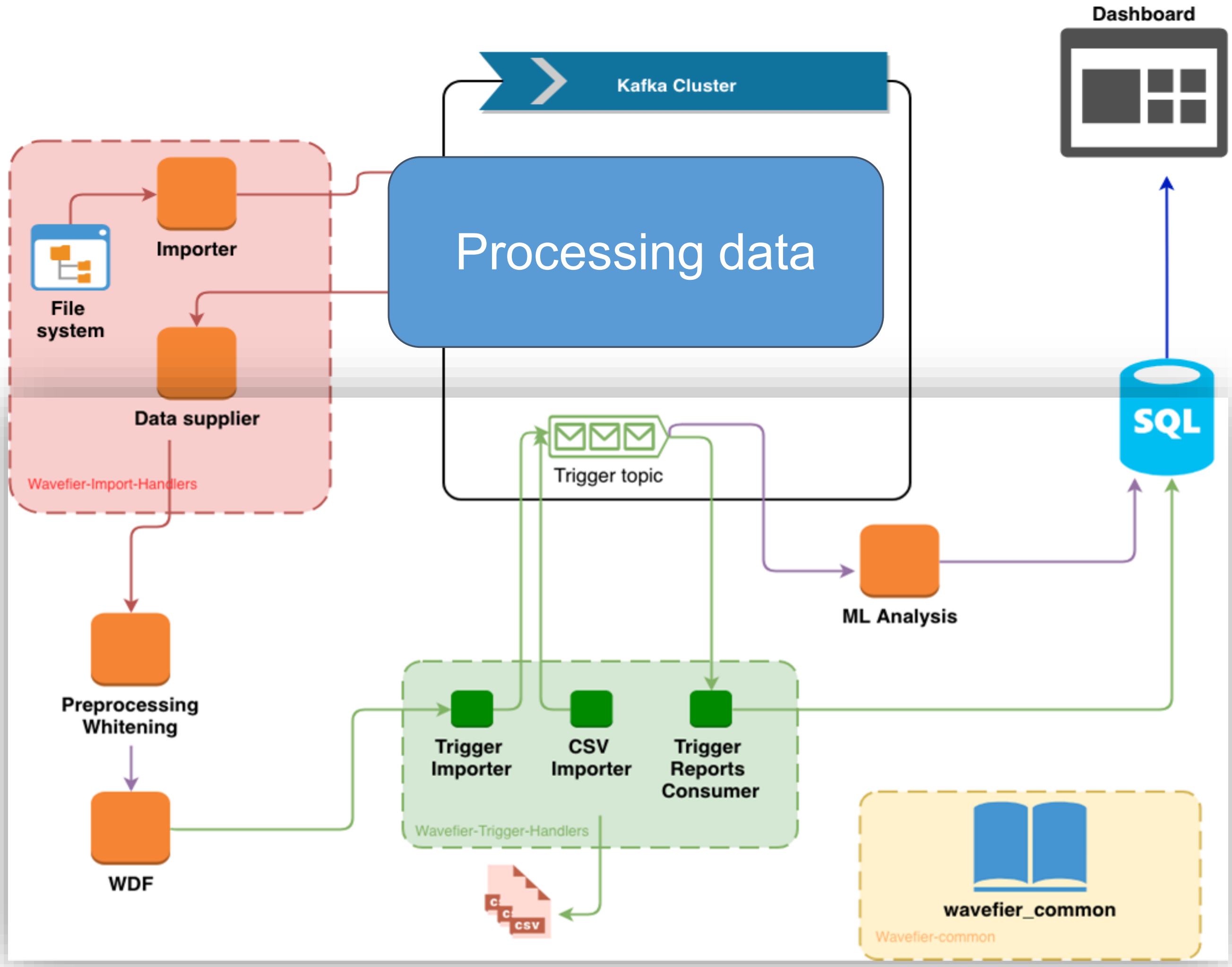
Offline data - Architecture



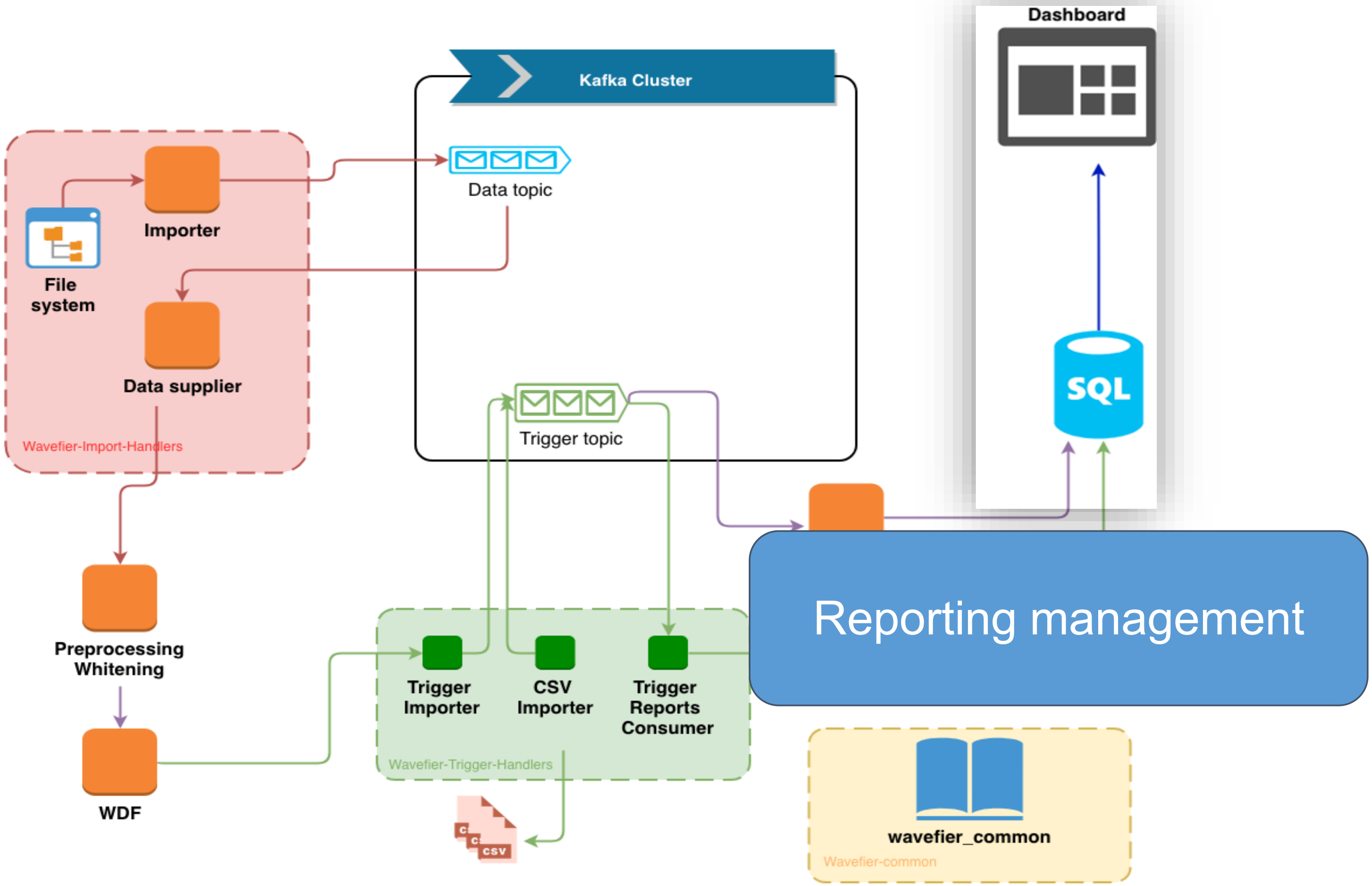
Offline data - Architecture



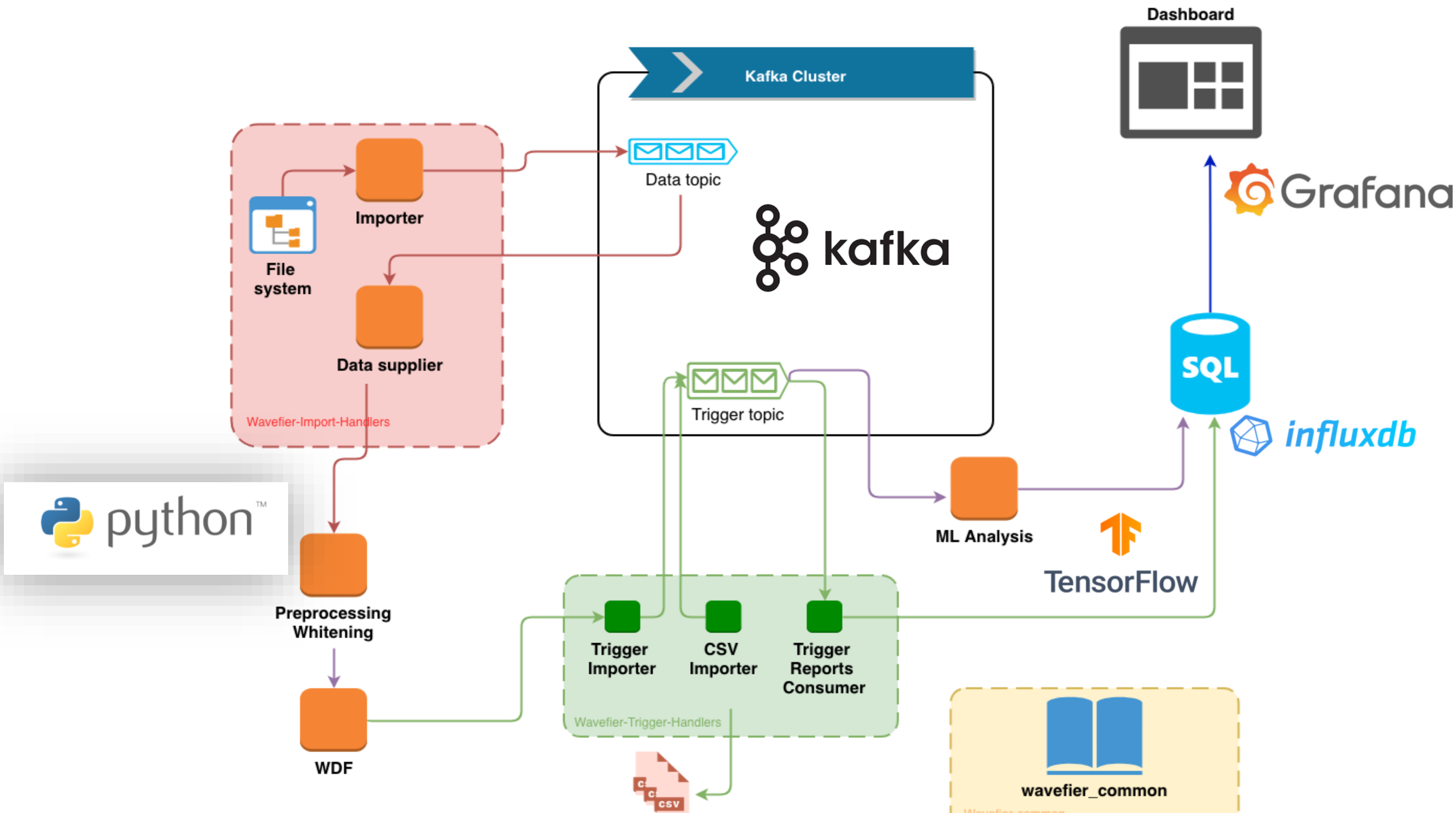
Offline data - Architecture



Offline data - Architecture



Offline data - Architecture



Apache Kafka

Open-source stream-processing software platform developed by LinkedIn and donated to the Apache Software Foundation

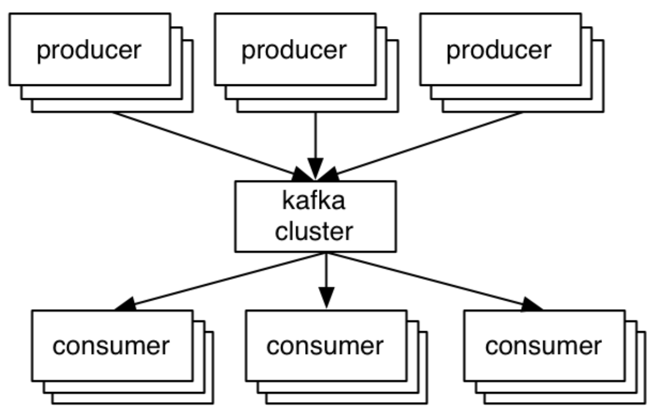
Apache Kafka® is a distributed streaming platform. What exactly does that mean?

A streaming platform has three key capabilities:

- Publish and subscribe to streams of records, similar to a message queue
- Store streams of records in a fault-tolerant durable way.
- Process streams of records as they occur.

Kafka is generally used for two broad classes of applications:

- Building real-time streaming data pipelines that reliably get data
- Building real-time streaming applications that transform or react to the streams of data



More info on:

https://kafka.apache.org

Grafana

Why grafana

- Useful build-in features
  - Authentication, Organization and user settings

- Mixed Datasource, Mix different data sources in the same graph
  - Grafana supports dozens of databases, natively. Mix them together in the same Dashboard.

- Native Notification and Alerting

InfluxDB

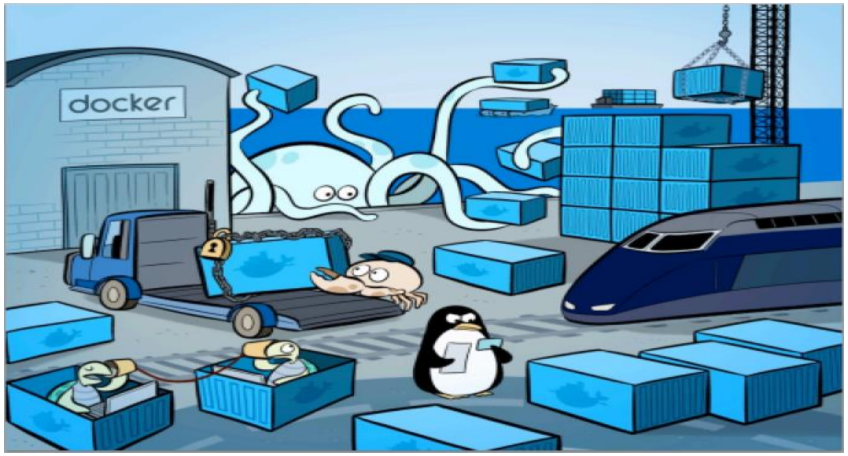
Why InfluxDB?

- Specific for time series database (TSDB)
  - All is designed as time series
- Friendly because InfluxDB have a SQL-like query language for interacting with it
- Grafana has Native support for InfluxDB

Architecture Deploy

Docker

- Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications.
- Docker take the concept of container and build an ecosystem around it that would simplify its use



Key benefits of Docker Containers

Hardware independent → moving everywhere

Speed

- No OS to boot = applications online in seconds

Portability

- Less dependencies between process layers = ability to move between infrastructure

Efficiency

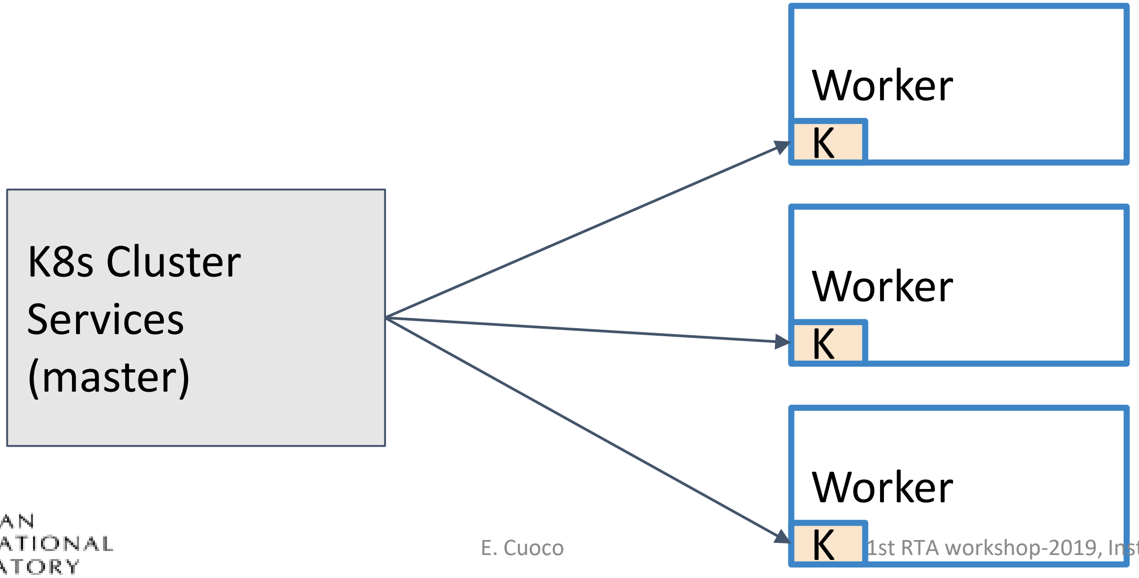
- Less OS overhead
- Improved VM density

Kubernetes (K8s)

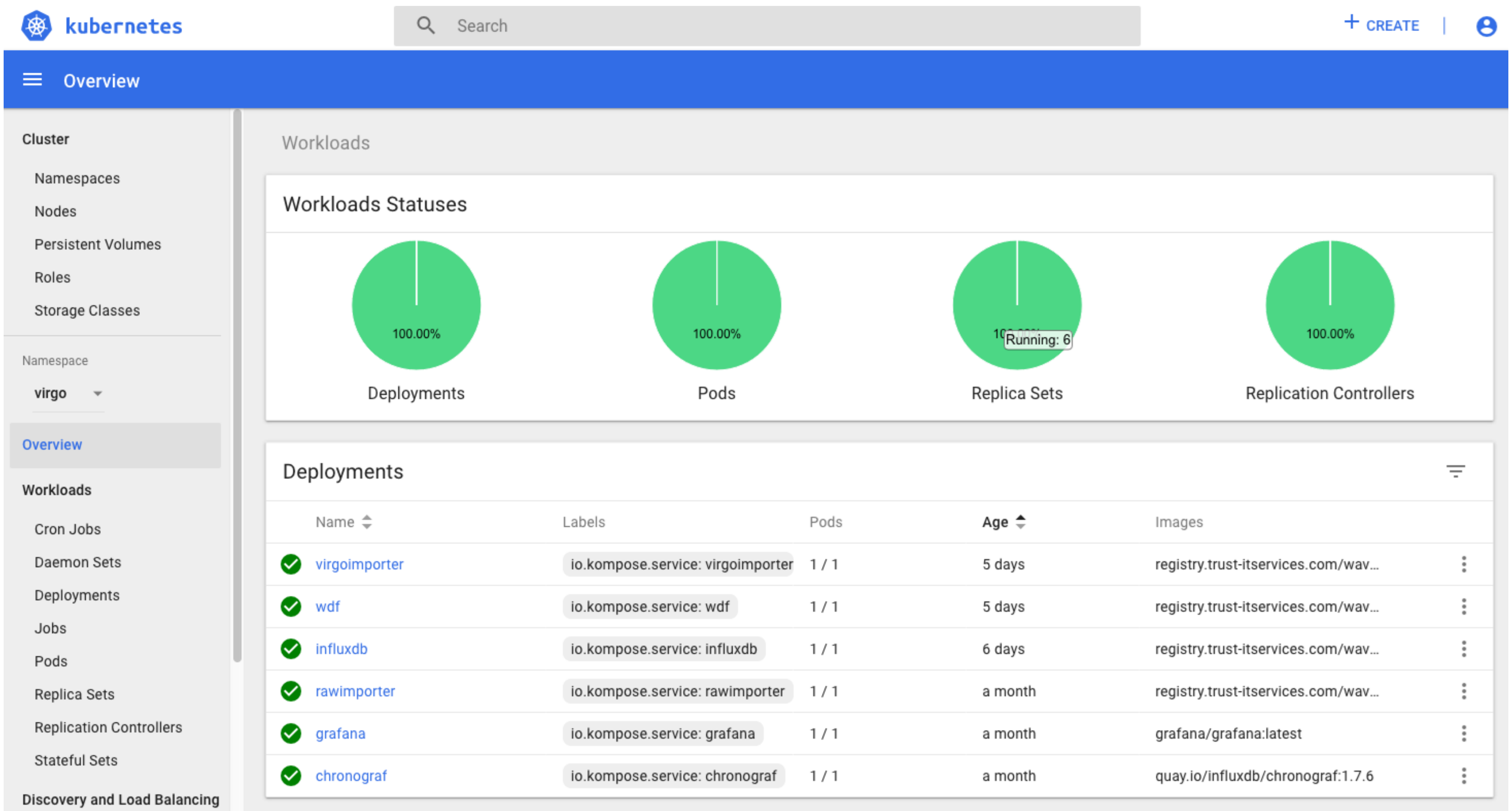
was a project spun out of Google as a open source next-gen container scheduler, designed as a loosely coupled collection of components centered around deploying, maintaining, and scaling applications.

Architecture overview

- Kubernets abstracts away the underlying hardware of the nodes and **provides a uniform interface for applications** to be both deployed and consume the shared pool of resources.
- Masters: are responsible at a minimum for running the API Server, scheduler, and cluster controller. They commonly also manage storing cluster state, cloud-provider specific components and other cluster essential services.
- Nodes: Are the ‘workers’ of a Kubernetes cluster. They run a minimal agent that manages the node itself, and are tasked with executing workloads as designated by the master.

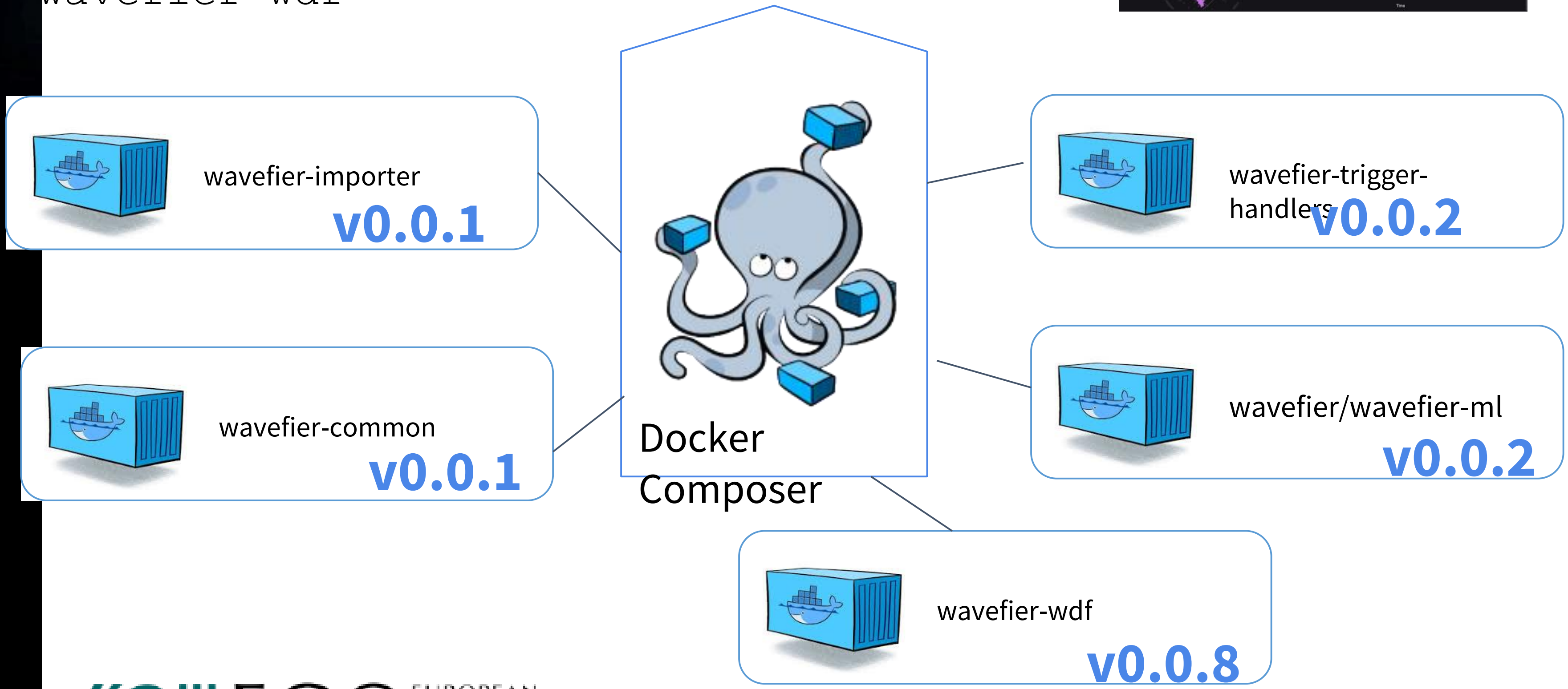


WaveFier running on Kubernetes



Same Software on Local Deployment

```
#> docker-compose up
wavefier-wdf
```



Software Management

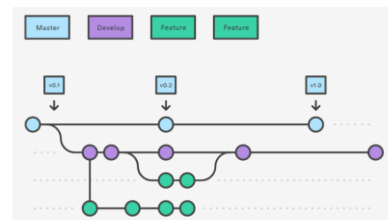
Why defines software management?

- Distributed Team (2 places)
  - Trust-it, EGO
- Different expertise
  - Physics, Software Engineer and Computer science
- One unique target

How we managed the software

1. Version of the software with common rules of release

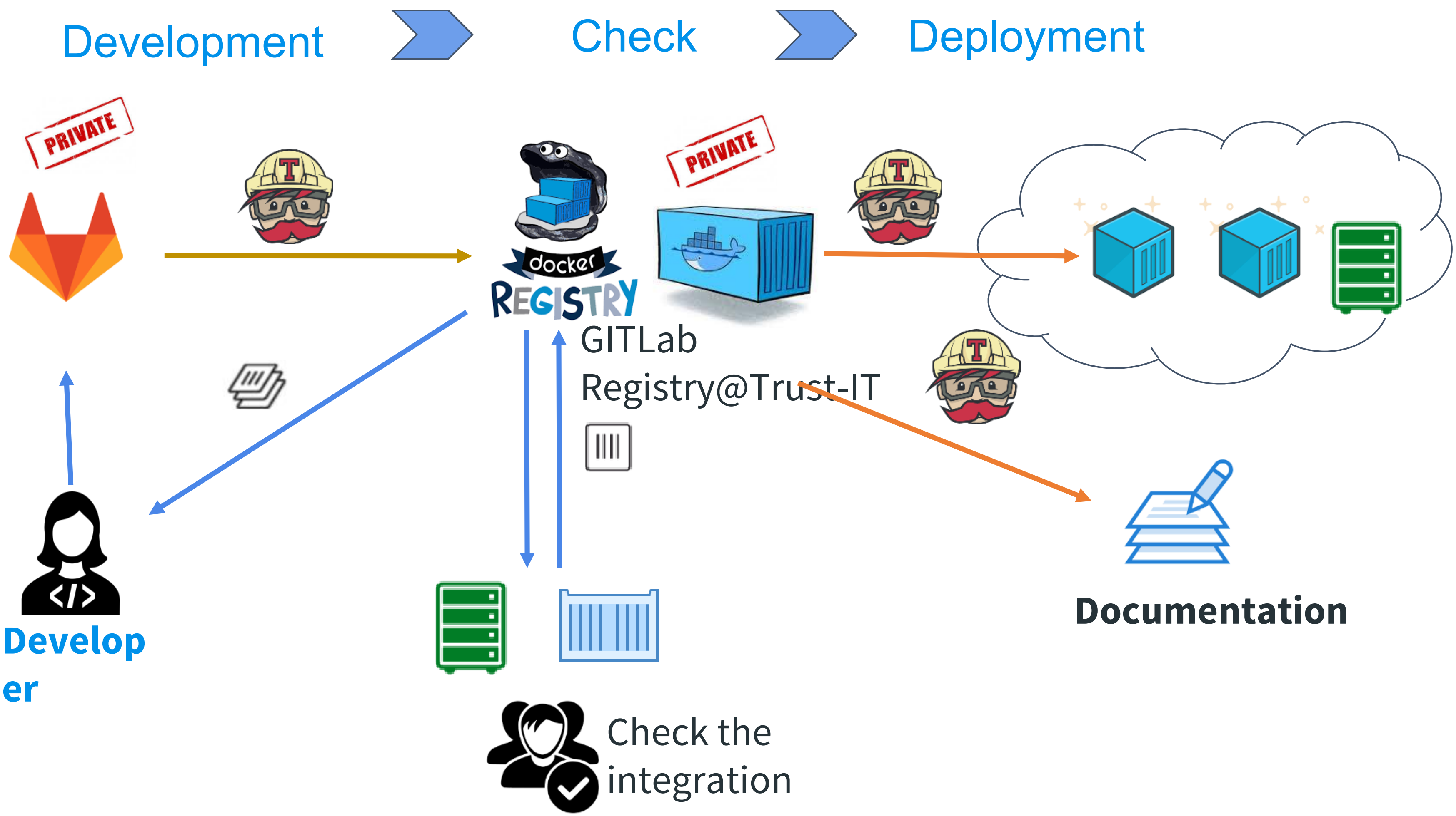
2. Setup Continuous Integration

git + 

3. Create as much automation as possible!

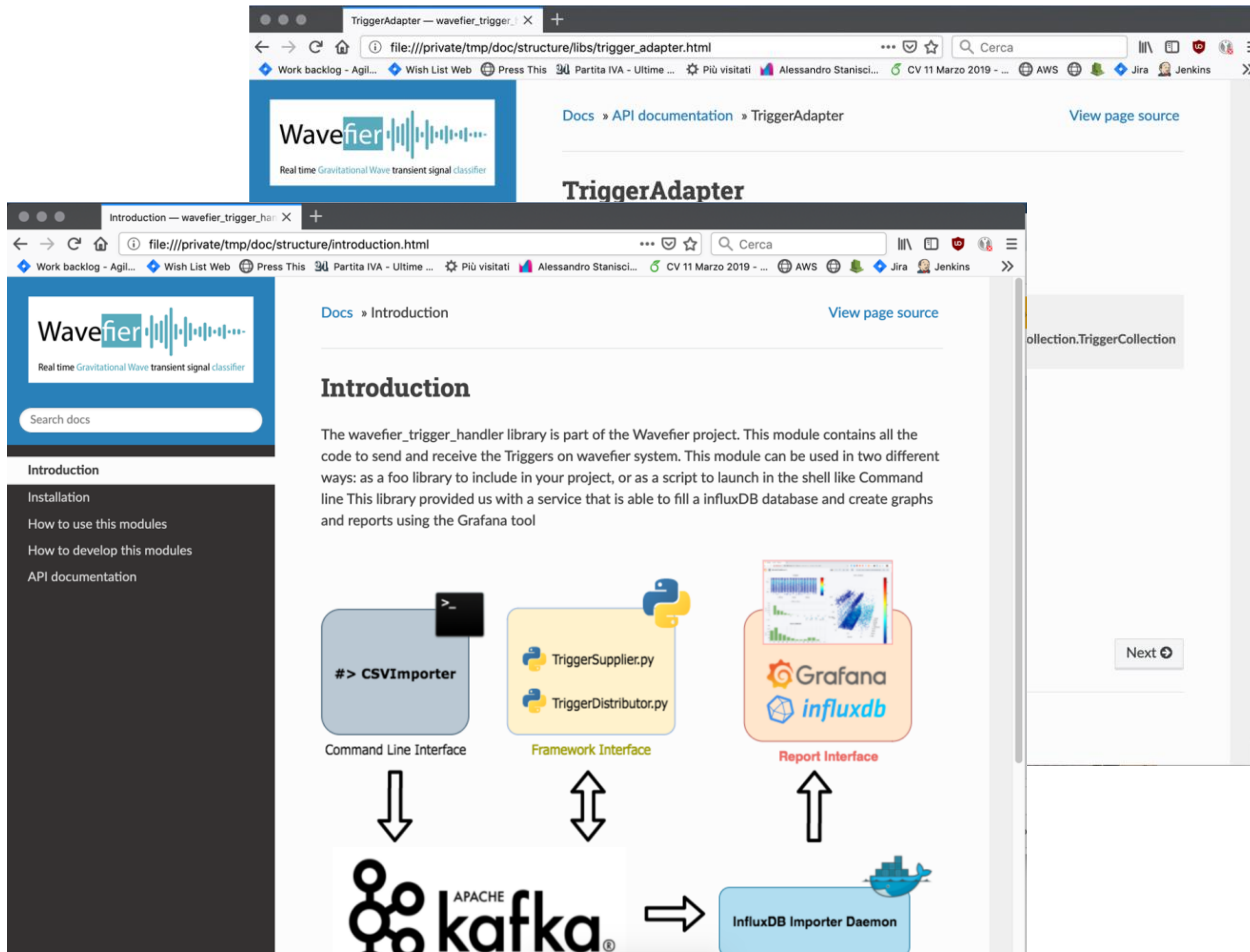
Travis CI

Automation anContinuous Integration



Automation and CI *on Wavefier*

The Documentation is also generated foreach git Commit



Data Management

Offline data vs Online data

Offline data

- Pick-up interferometer data
- Store data in files
- Access to cluster
- Move data in cluster

Online data

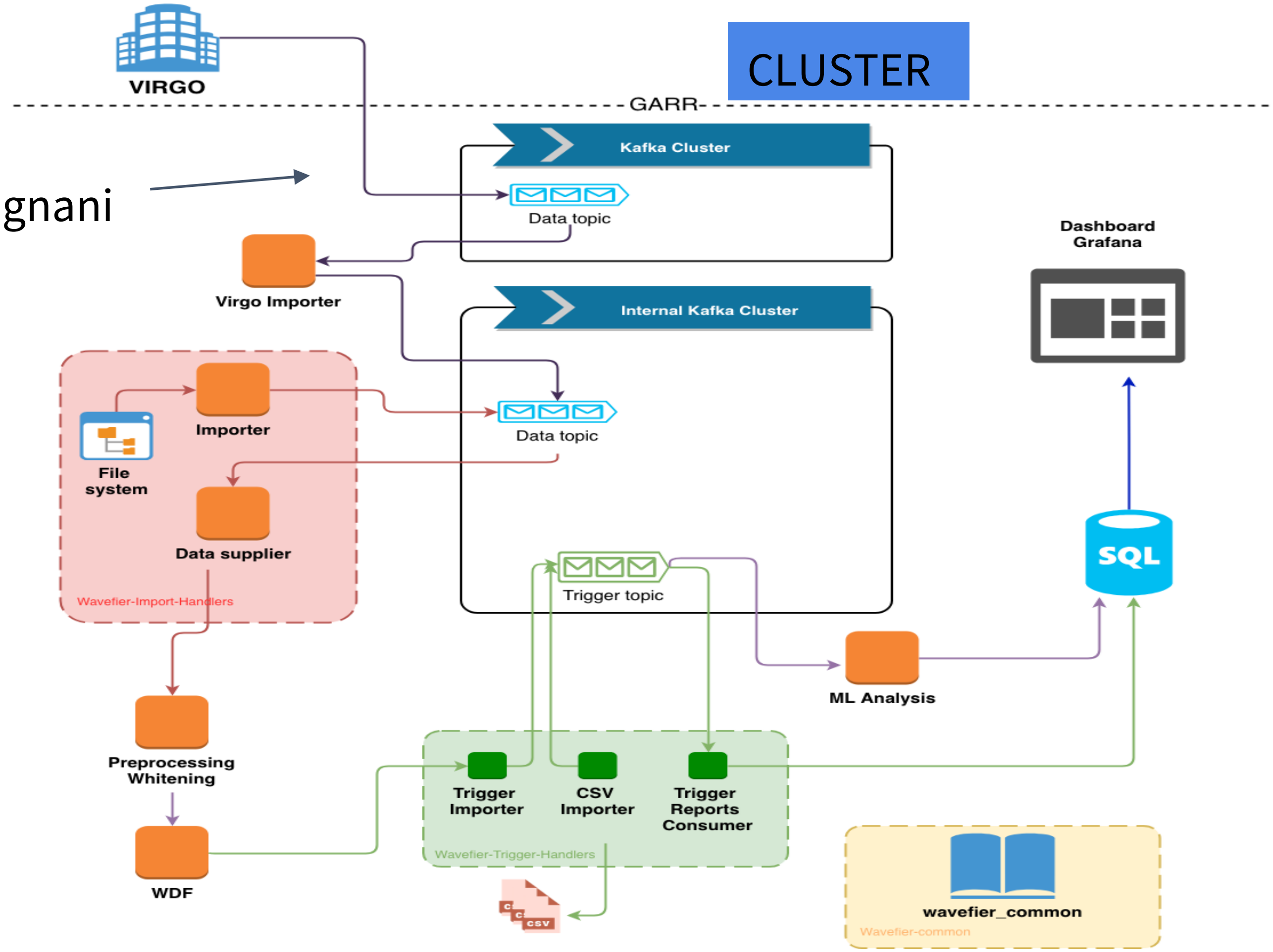
- Receiving data from different sources

Project Goal

Process streaming data

Online data - Architecture

Thanks to
Franco Carbognani



Current Grafana Dashboard with Classification Results

Video Demo


Wavefier - RTA v3.2
Channel None
Interval 10m
Add Filter:
Glitchgram - 10m
Trigger Time Distribution
No data points
SNR distribution - 10m
No data points
Frequency distribution - 10m
No data points
Duration-Frequency-Snr
CNN-1D classifier
Frequency
Time
Count

CREDITS:

LAPP and CNRS: Giovanni Lamanna, Jayesh Wagh

Trust-IT: Silvana Muscella, Emanuel Marzini, Filip Morawski, Alessandro Petrocelli, Alessandro Staniscia

and

Many thanks to GARR staff for their support: Giuseppe Attardi (coordinator dip. Cloud GARR), **Alberto Colla**, Alex Barchiesi, Claudio Pisa, Fulvio Galeazzi, Roberto di Lallo

Next step?

- Move from prototype/development to production
- Release the global documentation: from installation to user interface.
- Run Wavefier on cluster with Virgo on line data
- Investigate the use of much more sophisticated Machine/Deep learning algorithm, using GPU
- Upload the code in Asterics/Obelics catalogue and move further in ESCAPE project

ESCAPE

European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

ASTRONOMY & PARTICLE PHYSICS CLUSTER

Project Coordinator: Giovanni LAMANNA

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Background

- ESCAPE is based on the capacity building of the H2020 ASTERICS cluster of ESFRI projects (in astrophysics and astroparticle physics) addressing Big Data challenges and already succeeding in:
  - enabling interoperability between the facilities,
  - minimising fragmentation,
  - encouraging cross-fertilisation and
  - developing joint multi-messenger capabilities.

Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477

Funded by the European Union's Horizon 2020 - Grant N° 824064

Domain Cluster projects

H2020-INFRAEOSC-04-2018 call

Clusters to ensure the connection of the EFRI RIs with EOSC (and the construction of EOSC)

Expected impact:

- *Improve access to data and tools leading to new insights and innovation*
- *Facilitate access of researchers to data and resources for data driven science.*
- *Create a cross-border open innovation environment.*
- *Rise the efficiency and productivity of researchers through open data services and infrastructures for discovering, accessing, and reusing data.*
- *Foster the establishment of global standards.*
- *Develop synergies and complementarity between involved research infrastructures.*
- *Adopt common approaches to the data management for economies of scale.*

Funded by the European Union's Horizon 2020 - Grant N° 824064

ESCAPE work programme

WP1 MIND. Leader: Giovanni Lamanna, LAPP-CNRS
Management and policy.

WP2 DIOS. Leader: Simone Campana, CERN
Contribute to the federation of global EOSC resources through an implementation of the Data-Lake concept (evolution of WLCG and other ESFRI RIs computing models) to manage extremely large volumes of data up to the multi-exabyte scale

WP3 OSSR. Leader: Kay Graf, FAU
Support for "scientific software" as a major component of the ESFR-RI “data” to be stored and displayed in EOSC via dedicated community-based catalogues. Implementation of a community-based approach for the continuous development of shared software and for training of researchers and data scientists.

WP4 CEVO. Leader: Mark Allen, CDS-CNRS
Extend FAIR standards, methods, tools of the Virtual Observatory to a broader scientific context; demonstrate EOSC's ability to include existing platforms.

WP5 ESAP. Leader: Michiel van Haarlem, ASTRON-NWO
Implementation of scientific analysis platforms enabling EOSC researchers to organize data collections, analyse them, access ESFRI's software tools, and provide their own customized workflows.

WP6 ECO. Leader: Stephen Serjeant, Oxford Open University
Citizen Science, Open Science et Communication

Funded by the European Union’s Horizon 2020 - Grant N° 824064

Task 3.4 description

- Task 3.4:
  Foundation of Competence for Software and Service Innovation (COSSI)
  - Lead: Elena Cuoco (EGO)
  - Partners: AIP, CNRS- LAPP, NWO-I- CWI, **EGO**, HITS, INFN, OROBIX, UNITOV
  - Activities and Aims:
    - Review and further develop new approaches and developments for data exploitation
    - Starting with machine and deep learning
    - Establish innovation competence group

Funded by the European Union's Horizon 2020 - Grant N° 824064

Let’s play with tutorial on GW classification

Thank you