Strategy for Classification of Noise Transients in Advanced Detectors

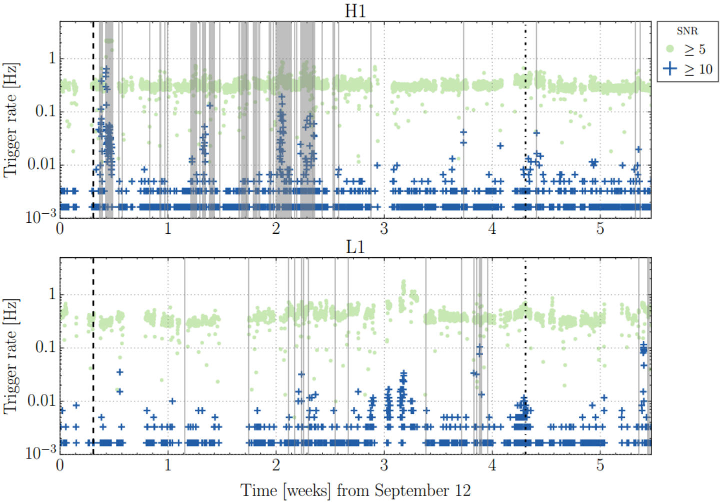
Elena Cuoco[1], J.Powell[2], A.Torres-Fornè[3], R.Lynch[4], D.Trifiró[5], M.Cavaglià[6], Ik S. Heng[2], J.Font[3] [7]

[1]EGO and PISA INFN, [2]SUPA and IGR Glasgow, [3]Universitat de València, [4]MIT, [5]Università di Pisa,[6]University of Mississippi, [7]Observatori Astronòmic, Universitat de València

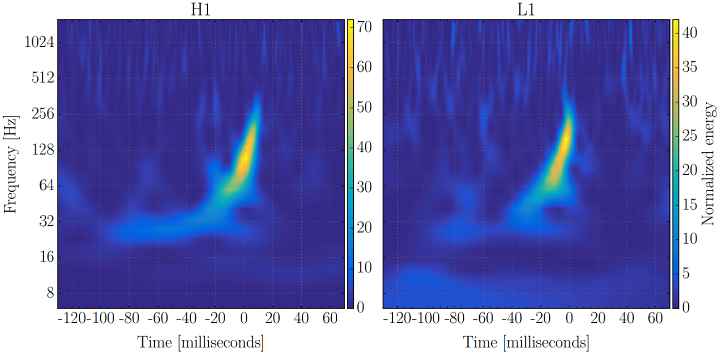
September 12, 2016

VIR-0391A-16

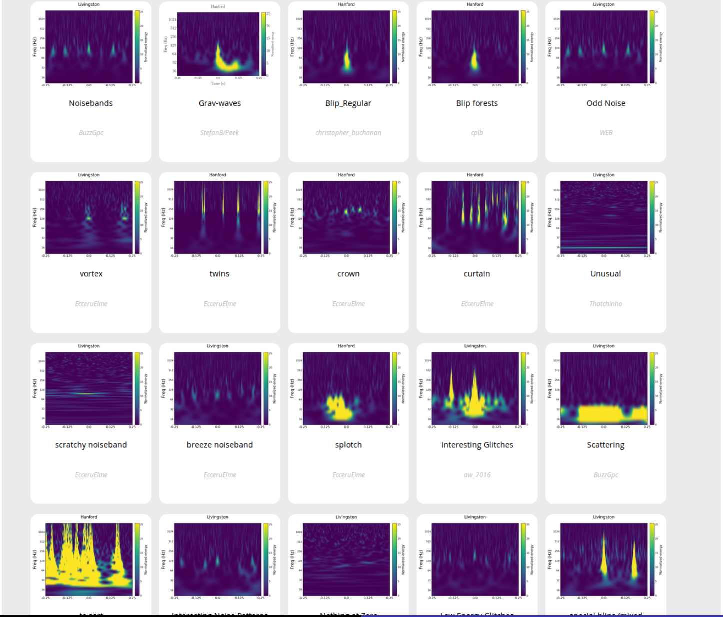
Typical glitchgram for detectors



Our 'typical' gravitational waves



Our glitch zoo



Noiseband

BuzzGpc

Grav-waves

StefanB/Peek

Blip_Regular

christopher_buchanan

Blip forests

cplb

Odd Noise

WEB

vortex

EcceruElme

twins

EcceruElme

crown

EcceruElme

curtain

EcceruElme

Unusual

Thatchinho

scratchy noiseband

EcceruElme

breeze noiseband

EcceruElme

splotch

EcceruElme

Interesting Glitches

aw_2016

Scattering

BuzzGpc

to sort

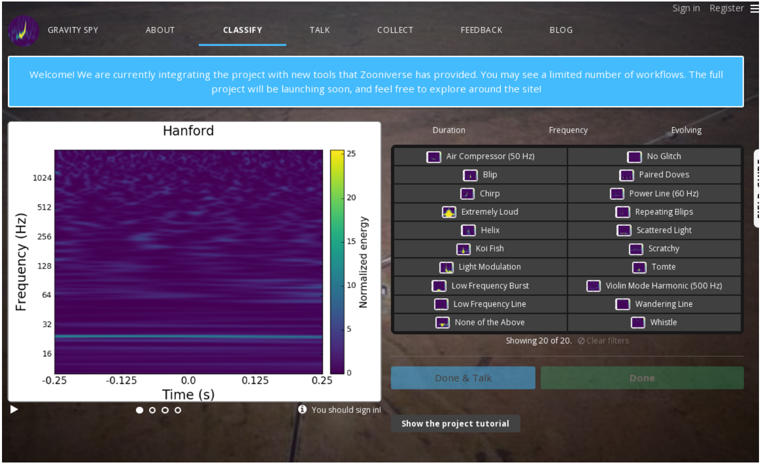
Interesting Noise Patterns

Nothing at Zero

Low Energy Glitches

special blips (mixed

GravitySpy project



Welcome! We are currently integrating the project with new tools that Zooniverse has provided. You may see a limited number of workflows. The full project will be launching soon, and feel free to explore around the site!

Air Compressor (50 Hz)	No Glitch
Blip	Paired Doves
Chirp	Power Line (60 Hz)
Extremely Loud	Repeating Blips
Helix	Scattered Light
Koi Fish	Scratchy
Light Modulation	Tomte
Low Frequency Burst	Violin Mode Harmonic (500 Hz)
Low Frequency Line	Wandering Line
None of the Above	Whistle

Showing 20 of 20. Clear filters

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

Why Glitch Classification?

- As prompt characterization of noise will be critical for improving sensitivity, a fast method for glitch classification was needed.
- The detchar group proposed a challenge for the development of a method for automatic classification of glitches.
- We present three methods developed for automatic glitch classification.
- We started using simulated data sets to better understand the performance of the different glitch classifying codes.
- We tested our pipelines on LIGO ER7 data

Principal Components.

- All three methods use at some stage Principal Components (PCs).
- PCs are a set of orthogonal basis vectors, which are ordered so that the first PC represents the most common feature of a set of waveforms.
- Therefore, a few PCs can be used to represent all the common features of the waveforms.
- Each waveform m can be used to create a matrix A where each column of A corresponds to one of the waveforms of length n.
- The nxm waveform matrix A is factored so that

  $$A = U\Sigma V^T \tag{1}$$

  where V is $A^T A$ and Σ contains the eigenvalues.
- U contains the PCs, ranked by the eigenvalues, where the first PC represents the most common feature of the waveforms, the second PC represents the second most common feature and so on.

Principal Components.

- The signal model consists of a linear combination of PCs

$$\hat{h}_i = \sum_{j=1}^{k} U_j \beta_j \tag{2}$$

where $\hat{h}_i$ is the signal model, U_j is the jth PC from the U matrix and β_j is the corresponding PC coefficient.

- The values for β can be calculated by taking the dot product of h and U.

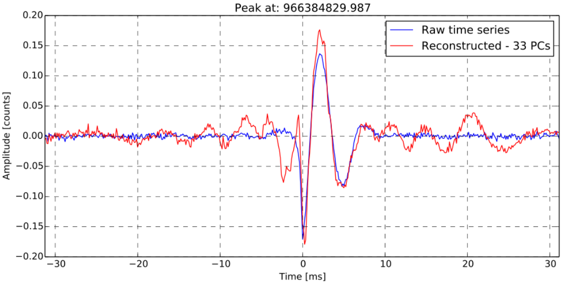
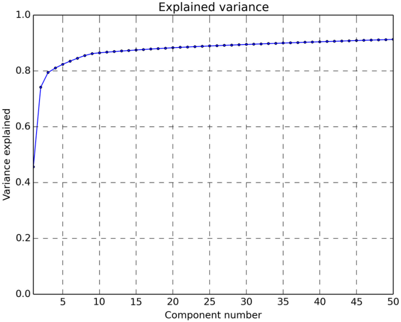


Figure: A glitch reconstructed by PCAT using 33 PCs.

Choosing the number of Principal Components.

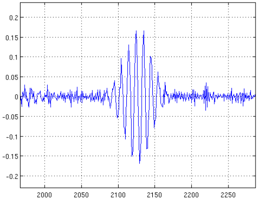
- Results can be strongly effected by the number of Principal Components.
- We use the variance method to choose the ideal number of Principal Components.

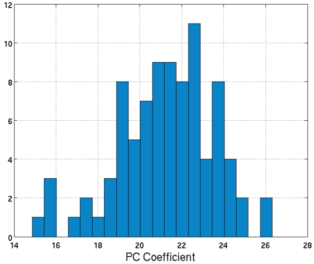


PC-LIB

- PC-LIB is an adaptation of the parameter estimation and model selection tool LALInference.
- A set of Principal Components for a type of glitch is made using the high pass filtered time series of fifty glitches for that type.
- A linear combination of the PCs, multiplied by the PC coefficients, is then used as the new signal model in LIB for each different population of noise transient. The different signal models for each glitch population can then be used for Bayesian model selection, which can determine the type of each new noise transient that is detected in the data.

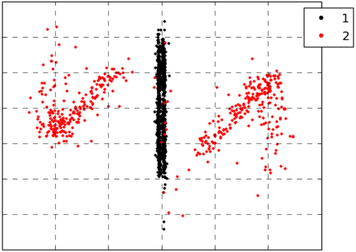
  For two competing models M_i and M_j the Bayes factor is given by the ratio of the evidences,
- Model selection can then be used to identify the correct glitch type.

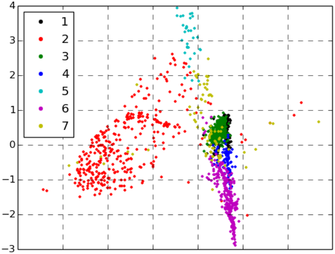




PCAT

- Principal Component Analysis for Transients (PCAT) is a python-based pipeline based on Principal Component Analysis.
- The time series of whitened glitches are stored in a matrix on which PCA is performed.
- PCAT uses the PC coefficients to classify the glitches by using a Gaussian Mixture Model (GMM) implementation of scikit-learn, which includes machine learning routines for model selection.It requires the user to specify the number of clusters and the number of principal components.
- The results of the PCA can be visualized with scatter plots of the principal component coefficients.





Wavelet Detection Filter (WDF): transform

The wavelet transform of a signal $f(t)$ is defined as

$$Wf(a,b) =< f, \psi_{a,b} >= \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{b}} \psi^*(\frac{t-a}{b}) \ \ dt \qquad (3)$$

where the base is a zero average function, centered around zero and with a finite energy. The entire base is obtained by translations and dilations of the base atom:

$$\psi_{ab}(t) \quad = \quad \frac{1}{\sqrt{b}} \psi(\frac{t-a}{b}) \qquad (4)$$

The wavelet transform has a time frequency resolution which depends on the scale b. Its time spread is proportional to b and its frequency spread is proportional to the inverse of b.

WDF-denoising

Let us consider a signal x_i which is corrupted by additive Gaussian random noise $n_i \sim N(0, \sigma^2)$ as follow

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

Let W be an orthogonal wavelet transform. If we apply it to the sequence of data x_i we obtain

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

Now let T be a wavelet thresholding function. Then the wavelet thresholding based de-noising scheme can be written

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

that is we first take the wavelet transform of our noisy signal and pass it through the thresholding function, then the output is inverted and wavelet transformed.

WDF-trigger

The wavelet coefficients contain the energy of the signal at different scale. After the wavelet thresholding, we selected the highest coefficients of the wavelet transform which are supposed to contain only the signal and not the noise.

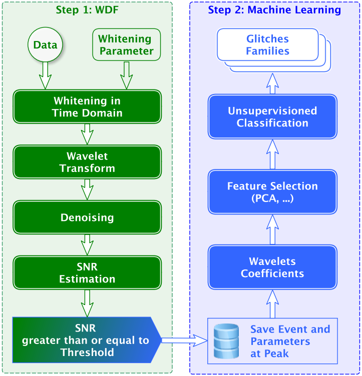
$$E_s = \sqrt{\sum_{k,j} w_{k,j}^2} \tag{5}$$

being $w_{k,j}$the wavelet coefficients above the threshold.
In this way $E_s$represent the signal energy content, so we can build our receiver detector which represents the signal to noise ratio, as

$$SNR = \frac{E_s}{\hat{\sigma}} \tag{6}$$

WDF-ML: Machine Learning step

- Completely unsupervised algorithms. No target function
- Wavelets coefficients and Meta data (SNR, Freq,Duration) represents our "features"
- Features selection uses PCA transform an Spectral embedding on 2 dimensions
- The Gaussian Mixture Model (GMM) machine learning classifier is then applied to the outputs of WDF for classification.



Step 1: WDF

Data

Whitening Parameter

Whitening in Time Domain

Wavelet Transform

Denoising

SNR Estimation

SNR greater than or equal to Threshold

Step 2: Machine Learning

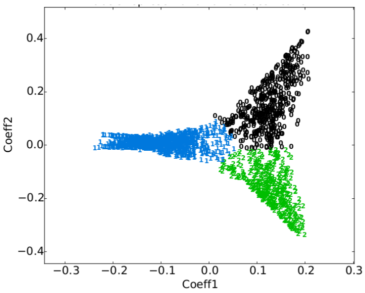
Save Event and Parameters at Peak

Wavelets Coefficients

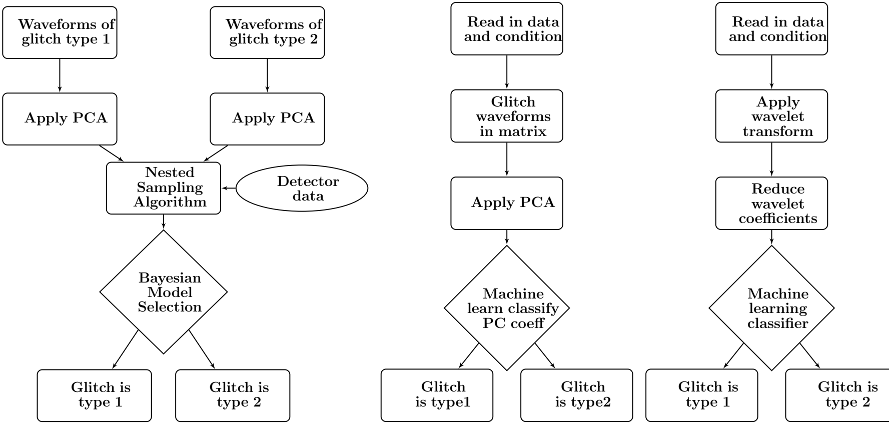
Feature Selection (PCA, ...)

Unsupervisioned Classification

Glitches Families

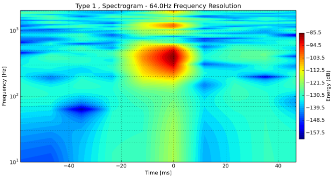


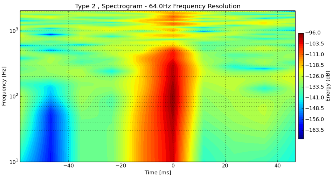
Pipelines summary: PC-LIB, PCAT,WDF-ML


Waveforms of glitch type 1
Waveforms of glitch type 2
Apply PCA
Apply PCA
Nested Sampling Algorithm
Detector data
Bayesian Model Selection
Glitch is type 1
Glitch is type 2
Read in data and condition
Glitch waveforms in matrix
Apply PCA
Machine learn classify PC coeff
Glitch is type1
Glitch is type2
Read in data and condition
Apply wavelet transform
Reduce wavelet coefficients
Machine learning classifier
Glitch is type 1
Glitch is type 2

MDC: Data set 1

- To test and compare methods we create a simulated data set in aLIGO Gaussian noise.
- Data set 1 is an ideal data set where all of the glitch types are well separated in frequency and SNR.
- The data set contains 1000 sine Gaussian waveforms and 1000 Gaussian waveforms in simulated Gaussian noise.
- The sine Gaussian waveforms have a frequency = 400Hz and an SNR between 5 and 30.
- The Gaussian waveforms are centred at f = 0Hz and have an SNR between 20 and 250.


Type 1 , Spectrogram - 64.0Hz Frequency Resolution


Type 2 , Spectrogram - 64.0Hz Frequency Resolution

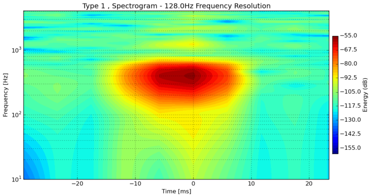
Data Set 1 Results

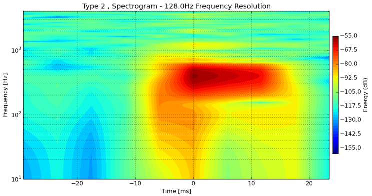
- Table shows the % of detected transients that were classified in each type.
- A few low frequency SG, and low SNR G were in the incorrect classes.
- Overall classification efficiency very good!

	SG	G
PCAT Type 1	99%	0%
PCAT Type 2	1%	100%
LIB Type 1	99.9%	5%
LIB Type 2	0.1%	95%
WDF Type 0	99.5%	2.4%
WDF Type 1	0.3%	46.1%
WDF Type 2	0.2%	51.5%

MDC: Data set 2

- We use a second data set to see if we can classify glitches by waveform morphology only.
- We use 1000 sine Gaussian waveforms and 1000 Ring-down waveforms.
- All waveforms have identical frequency 400Hz and a identical duration 2ms.
- The SNR of the simulated glitches is between 10 and 500.





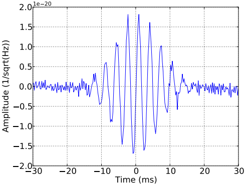
Data Set 2 Results

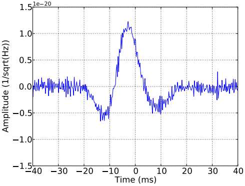
- Table shows the % of detected transients that were classified in each type.
- The few transients in the incorrect class are those with the lowest SNR.
- 5PCs PCAT, 7PCs LIB and 10 PCs WDF-ML.
- All methods can classify by waveform morphology alone.

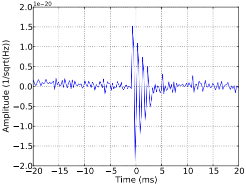
	SG	RD
PCAT Type 1	1.1%	97.4%
PCAT Type 2	98.9%	2.5%
LIB Type 1	97.8%	4.8%
LIB Type 2	2.2%	95.2%
WDF-ML Type 0	8.7%	100%
WDF-ML Type 1	48.0%	0%
WDF-ML Type 2	43.3%	0%

MDC: Data Set 3

- The third data set is to see what happens if different types have a very wide range of parameters.
- The simulated glitches are Gaussian, sine Gaussian and Ring-down waveforms at five second intervals.
- The frequencies are distributed linearly between 40-1500 Hz.
- Majority of the glitches have an SNR between 1 and 300.







Data Set 3 Results

- PCAT 20PCs, LIB 5PCs, WDF-ML 10PCs.
- All methods have the Gaussians in there own class.
- Cannot distinguish between the sine Gaussian and Ring-down waveforms when the parameter range is so large.

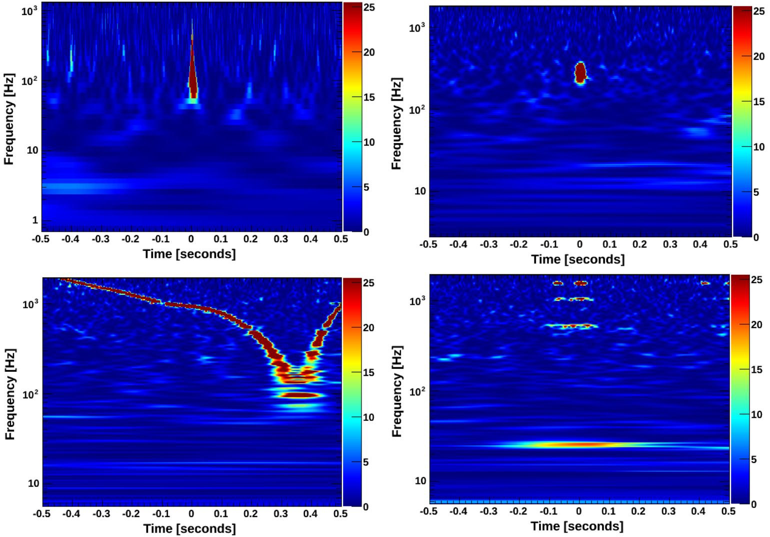
	SG	G	RD
PCAT Type 1	15.5%	0%	13.6%
PCAT Type 2	36.8%	0%	41.4%
PCAT Type 3	14.2%	0%	13.0%
PCAT Type 4	9.1%	0%	13.0%
PCAT Type 5	0.8%	0%	0.3%
PCAT Type 6	21.8%	0%	17.2%
PCAT Type 7	1.8%	100%	1.5%
LIB Type 1	39.5%	4.9%	23.8%
LIB Type 2	17.3%	88.3%	23.2%
LIB Type 3	43.3%	6.8%	53.0%
WDF-ML Type 0	89.5%	9.6%	86.9%
WDF-ML Type 1	5.9%	49.7%	7.0%
WDF-ML Type 2	4.6%	40.7%	6.1%

Classification methods for noise transients in advanced gravitational-wave detectors Class. Quant. Grav., 32 (21), pp. 215012, 2015.

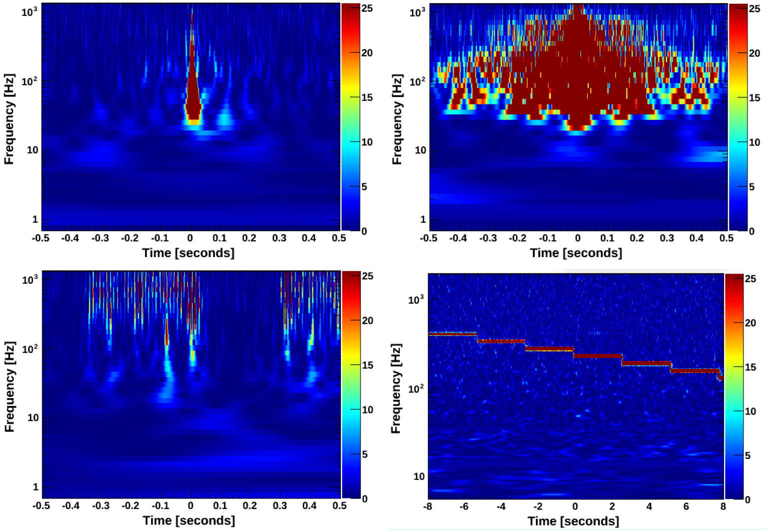
Real Data: ER7 LIGO data

- Data from the 7th aLIGO engineering run (ER7), which began on the 3rd of June 2015 and finished on the 14th of June 2015. The average binary neutron star inspiral range for both Hanford and Livingston detectors in data analysis mode during ER7 was $50 - 60$ Mpc.
- The total length of Livingston data analysed is ~ 87 hours.
- The total length of Hanford data analysed is ~ 141 hours.

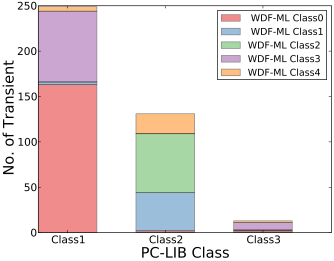
Real Data: ER7 L1

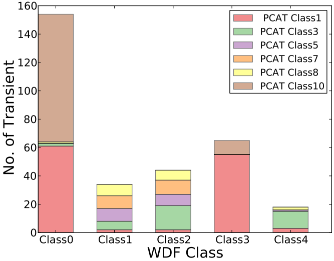


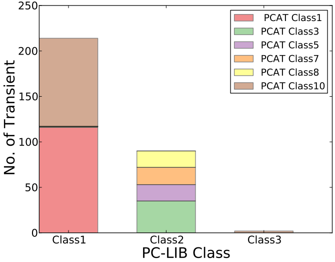
Real Data: ER7 H1



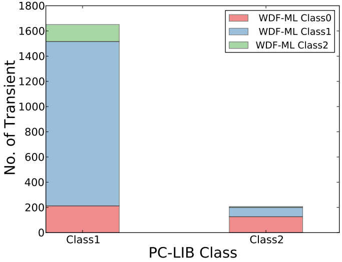
ER7 L1 results

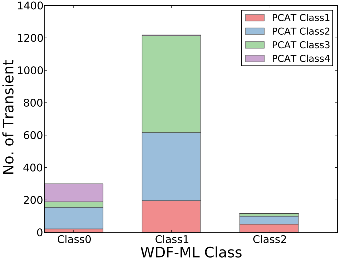


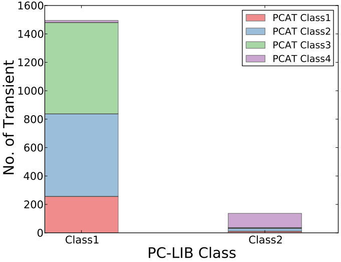




ER7 H1 results







Conclusion

- In the ER7 data from aLIGO Livingston PCAT missed 90 transients and classified 95% of the remaining transients correctly.
- PC-LIB missed 33 transients and classified 98% of the remaining transients correctly.
- WDF-ML classified all transients and 97% of them were correct.
- In aLIGO Hanford PCAT missed 120 transients and classified 99% of the remaining transients correctly.
- PC-LIB missed 6 transients and classified 95% of the remaining transients correctly.
- WDF-ML classified all transients and 92% of them were correct.
- We conclude that our methods have a high efficiency in real non-stationary and non-Gaussian detector noise.

To be submitted:
Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data. (by the authors)

What's next?

- Three different methods have been developed for the fast classification of noise transients.
- Transients are split in to types by waveform morphology first, and then can be split up in to further types by frequency and SNR.
- Results are similar for all methods.
- We plan to use Dictionary Based Algorithm
- We plan to use Images Deep Learning Classification
- Next we plan on looking at how these codes perform when using data from multiple auxiliary channels.