Glitch Classification for Gravitational Wave Interferometry using Machine Learning Techniques

-Rita Abani 19244
Department of Electrical Engineering and Computer Science, IISER-B

What are Gravitational Waves ?

Gravitational waves are 'ripples' in space-time caused by some of the most violent and energetic processes in the Universe.

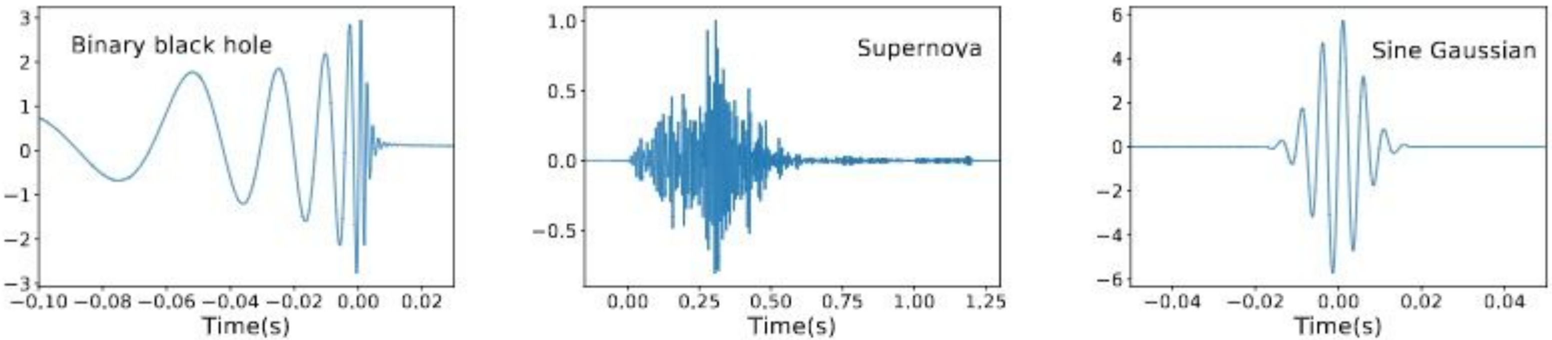
The strongest gravitational waves are produced by cataclysmic events such as colliding black holes, supernovae (massive stars exploding at the end of their lifetimes), and colliding neutron stars.

Researchers have relied on an object’s brightness as a rough gauge for its distance. But this approach carries endless complications. Gravitational waves help solve this dilemma and are a part of Einstein’s proof for GTR.

Why is removal of Glitches important for G-Wave data analysis.

Glitches are non gaussian noise transients that mimic true gravitational waves, they occur at different rates based on what is happening within the detector and in the environment around the detector. At their highest rate, they can occur at about **3 times per second in the detectors.**

Glitches also make our searches for gravitational waves less efficient, and generally make the **LIGO instruments less sensitive to astrophysical events.** It is important that we know as much about glitches as possible, in order to separate them from true signals, remove them from the data, or in the best case eliminate them from the detector entirely.



A little bit about the dataset

Real data from LIGO’s livingston and Hanford, labeled via the Gravity Spy catalogue.

Shape of the dataset = 6667 X 9

The following parameters are used to describe a glitch:

1. GPStime - GPS time of the first peak

2. peakFreq - frequency of the strongest peak of time series signal

3. snr - Signal to Noise Ratio of the glitch

4. centralFreq - central frequency of the signal

5. duration - duration of the glitch in the time series data

6. bandwidth - bandwidth of the signal

7. id - id of the glitch; this parameter is used to find a particular file with time-series in the `hdf5` directory

8. ifo - name of the interferometer; in the challenge we use H1 and L1 (Hanford and Livingston Interferometers)

9. label - type of the glitch

More about the dataset

```
metadata_df.shape
✓ 0.5s
(6667, 9)
```

```
metadata_df.isnull().sum()
✓ 0.9s
GPStime        0
peakFreq       0
snr            0
centralFreq    0
duration       0
bandwidth      0
id             0
ifo            0
label          0
dtype: int64
```

```
metadata_df.dtypes
✓ 0.5s
GPStime        float64
peakFreq       float64
snr            float64
centralFreq    float64
duration       float64
bandwidth      float64
id              object
ifo             object
label           object
dtype: object
```

```
metadata_df.duplicated()
✓ 0.1s
0       False
1       False
2       False
3       False
4       False
        ...
6662    False
6663    False
6664    False
6665    False
6666    False
Length: 6667, dtype: bool
```

	GPStime	peakFreq	snr	centralFreq	duration	bandwidth	id	ifo	label
0	1.134828e+09	32.246	40.137	1601.119	7.250	3183.373047	LYD73IJEbP	H1	Scattered_Light
1	1.134164e+09	29.897	38.256	2965.068	2.500	5894.235352	kQOi8X6807	H1	Scattered_Light
2	1.134478e+09	29.636	32.589	41.479	2.313	47.056068	wypbhS6TAa	H1	Scattered_Light
3	1.134478e+09	31.943	29.462	2532.243	3.000	5035.515137	aKFWKfLE8l	H1	Scattered_Light
4	1.137056e+09	32.552	28.537	1167.148	4.750	2324.383057	yWSM6mnbBM	H1	Scattered_Light

```
array(['Scattered_Light', 'Repeating_Blips', 'Violin_Mode', 'Power_Line',
       'Whistle', 'Scratchy', 'Helix', 'Light_Modulation',
       'Wandering_Line', 'Low_Frequency_Burst', 'Koi_Fish',
       'Low_Frequency_Lines', 'Blip', '1400Ripples', 'Chirp',
       'Extremely_Loud', 'None_of_the_Above', 'Paired_Doves', 'Tomte',
       'Air_Compressor', 'No_Glitch', '1080Lines'], dtype=object)
```

```
X_train.head()
```

✓ 0.6s

	peakFreq	snr	centralFreq	duration	bandwidth
717	31.263	24.589	2893.491	2.375	5742.492676
4267	10.737	14.756	14.194	3.250	16.102200
6055	39.717	12.456	51.401	0.625	58.312511
3459	137.711	10.344	250.227	0.031	395.537262
1019	137.711	70.589	2038.742	1.375	4048.512939

```
Y_train.head()
```

✓ 0.1s

```
717          Scattered_Light
4267     Low_Frequency_Lines
6055               No_Glitch
3459                    Blip
1019                Koi_Fish
Name: label, dtype: category
Categories (22, object): [1080Lines, 1400Ripples, Air_Compressor, Blip, ..., Tomte, Violin_Mode, Wandering_Line, Whistle]
```

What I plan on doing

Classify glitches using various classifiers listed below

Compare the following classifiers on the basis of the weighted avg of F-1 score from the classification report after tuning the parameters and then use the best model to plot results.

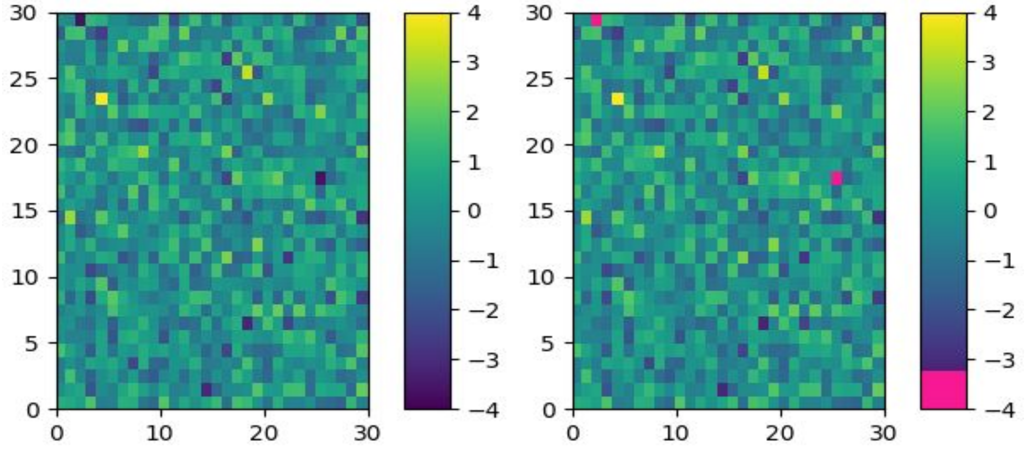
K-Nearest Neighbours Classifier	Support Vector Machine (Non linear kernel)
Random Forest	Decision Tree

How do I plan to visualize data ?

The matplotlib.colors.ListedColormap class is used to create colormap objects from a list of colors. This can be useful for directly indexing into colormap and it can also be used to create special colormaps for normal mapping.

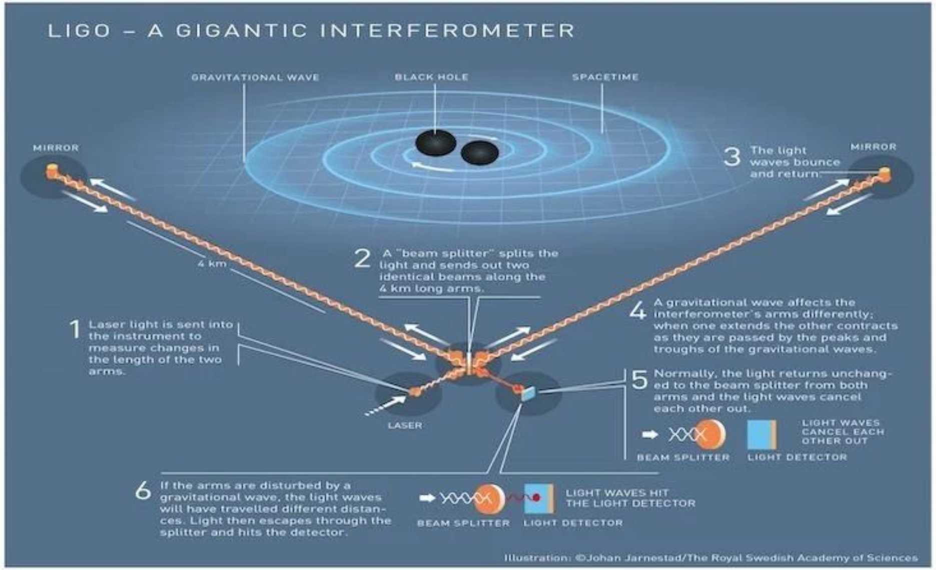
The listed colormap will show the predicted regions and training points of the best classifier

Read categories -> Turn them into digits -> Make a grid in the parameter space



How are the expected outcomes of use to the Scientific community

A robust ML based Glitch classifier can go a long way in reducing the dependence on heavy funding, costs and technicalities emanating from the technical requirements to build sophisticated Michelson Interferometers that filter glitches and environmental noises.



References

- https://www.nature.com/articles/d41586-018-04157-6
- https://www.birmingham.ac.uk/research/gravitational-wave/gravitational-waves-explained.aspx
- https://www.sciencedirect.com/topics/physics-and-astronomy/non-gaussian-noise
- https://www.zooniverse.org/projects/zooniverse/gravity-spy/about/faq
- https://discuss.analyticsvidhya.com/t/which-one-to-use-randomforest-vs-svm-vs-knn/2897/3