Identifying Glitches in Gravitational Waves Using Machine Learning

Name:	**Priya Shukla**
Registration No./Roll No.:	20214
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	January 12, 2023
Date of Submission:	April 16, 2023

1 Introduction

Gravitational waves are bends in the space-time continuum caused by violent interactions between massive, highly dense celestial bodies. Various factors, including cosmic activities such as supernovae, may cause glitches in these waves. Based on their wave-forms, they are classified into various categories such as Chirp, Blip, High Frequency, etc, to name a few.[1]

The data given, contains qualitative parameters that define the waveforms of the detected gravitational waves. These include bandwidth, peak frequency, snr, etc.

In this project, we develop suitable machine-learning techniques to identify and extract the features that differentiate various types of glitches. We build upon the preliminary ideas developed in Phase I of the project and come up with approaches towards data preprocessing and using a Normalization function.

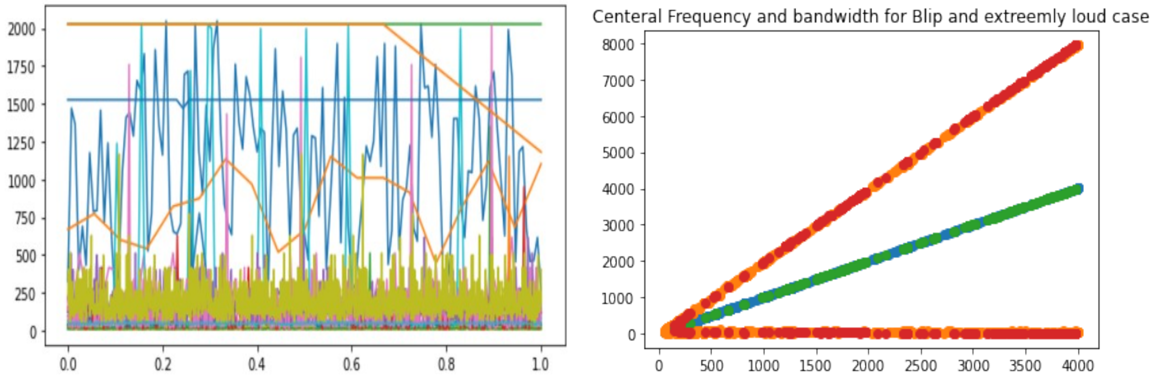


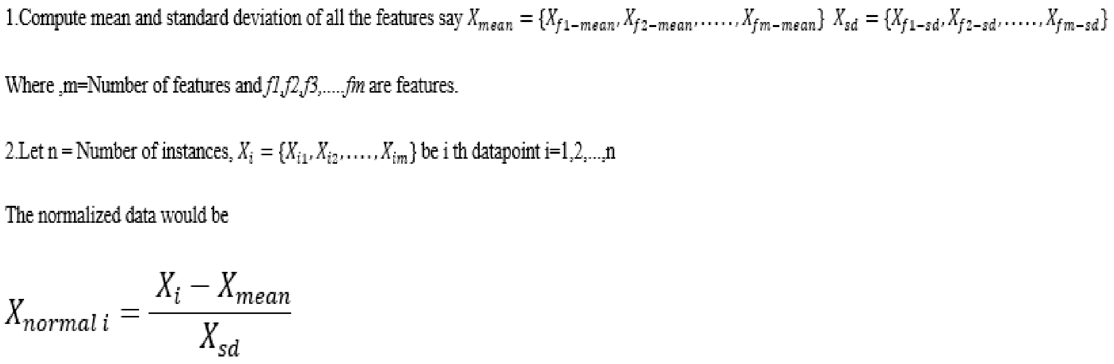
Figure 1: a: Peak frequencies for all classes b: Linear relation of central frequency and bandwidth

2 Methodology

We will be exploring the dataset visually as much as possible to infer as much as we can before applying our models. In this project we will be using a data normalization technique in order to train the model. The pseudo-code for this method is given as follows:

Next, we run the normalized data through various classifiers and experiment with the parameters of those classifiers to figure out the best loss function. We have used KNN, Decision tree [2]and Multinomial Logistic Regression[3], among others.

We will be comparing the results and illustrate our findings in the next section of this Project.



1.Compute mean and standard deviation of all the features say $X_{mean} = \{X_{f1-mean}, X_{f2-mean}, \ldots\ldots, X_{fm-mean}\}$ $X_{sd} = \{X_{f1-sd}, X_{f2-sd}, \ldots\ldots, X_{fm-sd}\}$

Where ,m=Number of features and *f1,f2,f3,.....fm* are features.

2.Let n = Number of instances, $X_i = \{X_{i1}, X_{i2}, \ldots\ldots, X_{im}\}$ be i th datapoint i=1,2,...,n

The normalized data would be

$$X_{normal\ i} = \frac{X_i - X_{mean}}{X_{sd}}$$

Figure 2: Pseudocode for Normalization routine

Apart from this, to address the problem of taking "ifo" (Interferometer location) into consideration for training the model, we run the models seperately on the overall dataset and then on the "H1" and "L1" subsets (Denoting the interferometer at Hanford and Livingston repectively).

We compare different models before and after hyper parameter tuning using GridSearchCV. The predicted class labels are given after training the Decision Tree model with optimum parameters as found outby using GridSearch

3 Experimental Analysis

Hyperlink for code.We have selected the most preferred model based on the f1 score. Random forest classifier was found to be of maximum test accuracy, but it is suspected of overfitting. The possible reason is that the data set is immensely biased towards a couple of features, and the Random Forest classifier is sensitive to such biases. We have thus deemed the next best, the Decision tree, as the classifier to use.

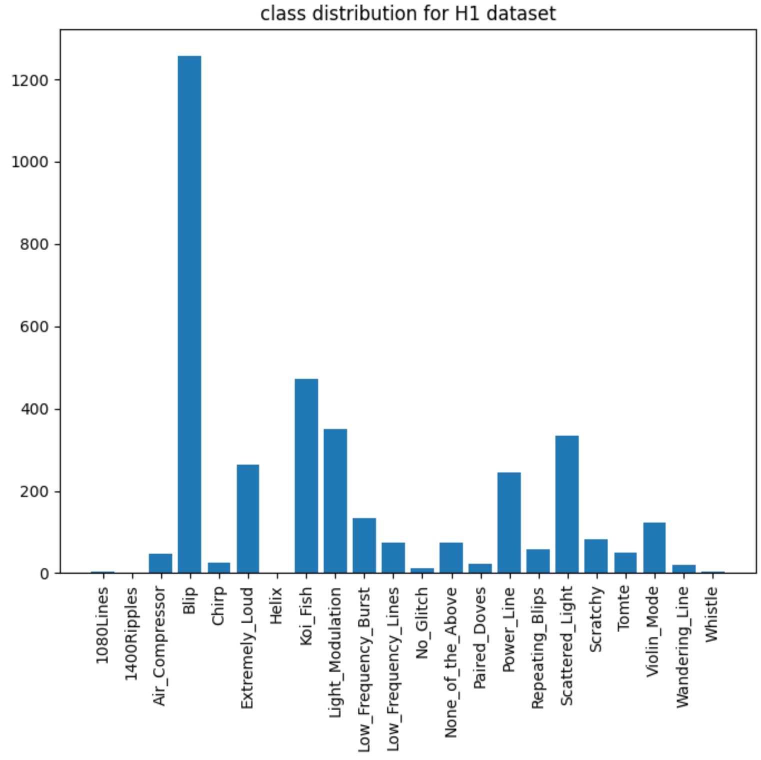
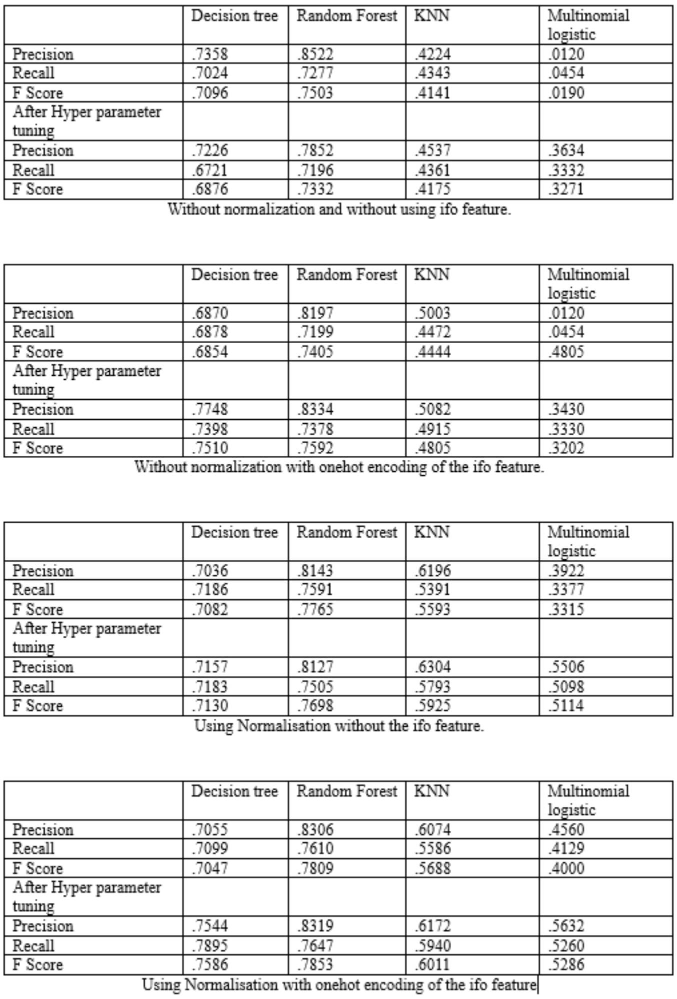


Figure 3: Class distribution for H1 and L1 dataset



	Decision tree	Random Forest	KNN	Multinomial logistic
Precision	.7358	.8522	.4224	.0120
Recall	.7024	.7277	.4343	.0454
F Score	.7096	.7503	.4141	.0190
After Hyper parameter tuning				
Precision	.7226	.7852	.4537	.3634
Recall	.6721	.7196	.4361	.3332
F Score	.6876	.7332	.4175	.3271

Without normalization and without using ifo feature.

	Decision tree	Random Forest	KNN	Multinomial logistic
Precision	.6870	.8197	.5003	.0120
Recall	.6878	.7199	.4472	.0454
F Score	.6854	.7405	.4444	.4805
After Hyper parameter tuning				
Precision	.7748	.8334	.5082	.3430
Recall	.7398	.7378	.4915	.3330
F Score	.7510	.7592	.4805	.3202

Without normalization with onehot encoding of the ifo feature.

	Decision tree	Random Forest	KNN	Multinomial logistic
Precision	.7036	.8143	.6196	.3922
Recall	.7186	.7591	.5391	.3377
F Score	.7082	.7765	.5593	.3315
After Hyper parameter tuning				
Precision	.7157	.8127	.6304	.5506
Recall	.7183	.7505	.5793	.5098
F Score	.7130	.7698	.5925	.5114

Using Normalisation without the ifo feature.

	Decision tree	Random Forest	KNN	Multinomial logistic
Precision	.7055	.8306	.6074	.4560
Recall	.7099	.7610	.5586	.4129
F Score	.7047	.7809	.5688	.4000
After Hyper parameter tuning				
Precision	.7544	.8319	.6172	.5632
Recall	.7895	.7647	.5940	.5260
F Score	.7586	.7853	.6011	.5286

Using Normalisation with onehot encoding of the ifo feature

Figure 4: Scores for different models under different conditions

4 Discussions and Plans

From the data given, one can infer how unbalanced the glitchs are. However the data should not be held responsible for inaccuracy of the machine learning model. Owing to rare occurences of gravitational waves themselves, it would be hard to expect accuracy for those glitches that occur even rarely (like ”1080Lines” glitch). Also, glitches that occur most often (Like ”Blip”) introduces some bias in the model. This makes the accuracy for less represented glitches feeble in comparision to those classes that are aptly represented.

One other thing that led us to make our models better at prediction was the normalisation function (and one hot encoding). One can clearly see the difference in results for the same test-train split on the same model but once where data is raw and once where data is normalized.

As part of our future plans, we plan to develop a novel train-test split method. The motivation is the unbalenced class data in the dataset. While some classes are represented in huge amounts (about half of the whole dataset), other are represented in small amounts (4 data points only). The new splitting method will be based on unbalanced splitting wherein the percentage of samples for each class will be different unlike the stratified split method (where percentage of data points for each in the training set is same)

Some exploration of the on-going cutting edge research in this field suggest that Deep Learning models[2] work best to classify the glitches. However the aim of this project was not to use Deep Learning algorithms and instead work with the Machine Learning algorithms.

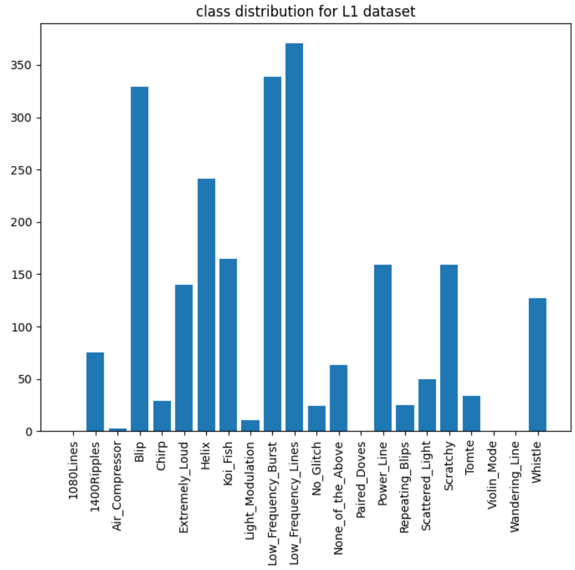


Figure 5: Class distribution for L1 dataset

References

[1] S Mukherjee. Classification of glitch waveforms in gravitational wave detector characterization. *Journal of Physics: Conference Series 243 (2010) 012006*, page 3, 2010.

[2] E. A. Huerta Daniel George, Hongyu Shen. Glitch Classification and Clustering for LIGO" with Deep Transfer Learning. *Workshop on Deep Learning for Physical Sciences (DLPS 2017), NIPS 2017, Long Beach, CA, USA*, 2017.

[3] Jason Brownlee. Multinomial Logistic Regression With Python. 2021.