2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 7-9, 2022

Glitch Classification for Gravitational Wave Interferometry Using Machine Learning

Rita Abani
Department of Electrical Engineering and Computer Science,
Indian Institute of Science Education and Research,
Bhopal
ritaabani22@gmail.com

2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT) | 978-1-6654-6855-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCAT55367.2022.9971811

***Abstract* - Gravitational waves are disturbances in the curvature of space-time generated by accelerated masses that propagate as waves outward from their source at the speed of light. A deep and analytical study of instrumental responses in the ecosystem of environmental noise is required for their optimal characterization. Hence of pertinent interest in this paper is the study of anomalous non Gaussian noise transients called 'Glitches'. These are short duration instrumental artifacts that can pollute the classification space. This paper tries to analyze appropriate machine learning techniques using a novel 'divide and conquer' based pipeline to classify glitches so that these can be subtracted from gravitational wave detector data for efficient detection and classification of gravitational wave ripples.**

Keywords - Gravitational waves, Divide and conquer algorithm, Machine Learning, Statistics, Data analysis, Data plotting and visualization, Inference, Causality, Correlation, Logic

I. Introduction

Gravitational waves are disturbances in the curvature of space-time generated by accelerated masses that propagate as waves outward from their source at the speed of light.The classification of glitches is essential owing to their high occurrence rates in LIGO data that often hazard and mimic true gravitational wave signals.

The data used in this project has been extracted from LIGO's Gravity Sky portal and contains metadata about these 'Glitches'. The train data contains information about the characteristics of a glitch like bandwidth, signal to noise ratio etc. (there are a total of 7 such features). Corresponding to the train data is a set of train labels or glitch labels which encompass 22 types of glitches along with their unique identification labels. In this project, various machine learning models from the sklearn or the scikit learn library in python namely K-nearest neighbours, Support Vector Machines, Random Forest and Decision Trees were used to train the data and develop an accurate model to classify glitches. That model was then run on the test data. In another sub-instance of this project, One Hot Encoding was used deal with the categorical variable 'ifo' or detector location and hence to target the research question 'Does the location of the interferometer have any impact on the classification of glitches'

The scatter plot in Figure 1 shows the correlation between various variables in the train data set.

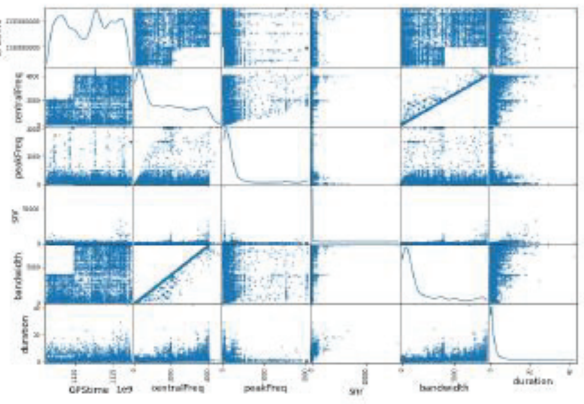


Fig. 1. correlation between various glitch features in the train data set

II. Methods

This project explored the use of ML models from sklearn like KNNs, SVMs, Decision Trees and Random Forest via a pipeline based on the 'Divide and Conquer Algorithm' juxtaposed to the existing practice of using a consolidated pipeline which is defined to run everything from the pre-processing to hyperparameter tuning to the evaluation and prediction as an entire atomic operation. A training routine was first designed, the pseudo-code of which is shown below in Figure 2. Feature selection was used to check the weighted contribution of the various columns in the data set:

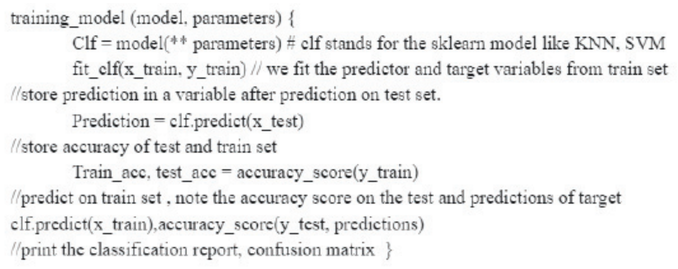
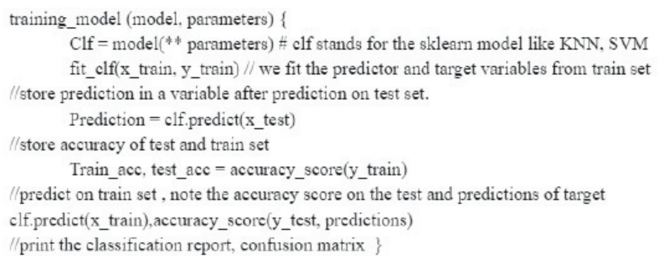

```
training_model (model, parameters) {
        Clf = model(** parameters) # clf stands for the sklearn model like KNN, SVM
        fit_clf(x_train, y_train) // we fit the predictor and target variables from train set
//store prediction in a variable after prediction on test set.
        Prediction = clf.predict(x_test)
//store accuracy of test and train set
        Train_acc, test_acc = accuracy_score(y_train)
//predict on train set , note the accuracy score on the test and predictions of target
clf.predict(x_train),accuracy_score(y_test, predictions)
//print the classification report, confusion matrix  }
```

Fig. 2. pseudo code for the training substructure

Before passing parameters to the training routing given above, hyperparameter tuning was separately performed using GridSearchCV as depicted by the pseudo code in Figure 3.



```
training_model (model, parameters) {
        Clf = model(** parameters) # clf stands for the sklearn model like KNN, SVM
        fit_clf(x_train, y_train) // we fit the predictor and target variables from train set
//store prediction in a variable after prediction on test set.
        Prediction = clf.predict(x_test)
//store accuracy of test and train set
        Train_acc, test_acc = accuracy_score(y_train)
//predict on train set , note the accuracy score on the test and predictions of target
clf.predict(x_train),accuracy_score(y_test, predictions)
//print the classification report, confusion matrix  }
```

Fig. 3. pseudo code for hyperparameter tuning

This divide and conquer approach of splitting the pipeline into a training routine and a parameter tuning routine reduced the time complexity of the code which is evident from the reduced time execution that was sufficed by a local intel i7 NVIDIA processor (without a GPU). For each of the ML models, i.e KNN, SVM, Decision Tree and Random Forest, parameter tuning was done followed by using the training routine with the subsequent optimal parameters. Two cases were analyzed; the first case involved performing classification of glitches without taking into account the effect of the only categorical or non-numeric variable 'ifo' that denotes the location of the interferometer (that can take only two values, 'L1' denoting Livingston and 'H1' denoting Hanford), the second case considered one hot encoding to convert the location (categorical variable into to numerical).

TABLE I. PERFORMANCE OF DIFFERENT CLASSIFIERS WITHOUT CONSIDERING LOCATION OF DETECTORS

Classifier	Precision	Recall	F-measure
KNN	0.66	0.66	0.66
SVM	0.69	0.69	0.67
Decision Tree	0.80	0.79	0.79
Random Forest	0.11	0.12	0.11

III. EVALUATION CRITERIA

Weighted F1 score , train and test accuracy were used to determine the best model for evaluation. The best model was run on the train samples and eventually unseen test data followed by a scatter plot that was made to visualize the results. In both the cases studied, the Decision tree algorithm was found to have the best F-measure. Subsequently, analysis involved more focus on the second case.

The Decision Tree classifier had the highest F-measure value along with an accuracy of 1 on the train data and 0.87 on test data. In scenario 2, after one hot encoding of 'ifo' , i.e converting the two locations, Hanford and Livingston into separate columns in the table, the best model was first implemented on data in case 'a') exclusively from Hanford , where 0.86, 0.87,0.86 was obtained as the precision, recall and weighted f measure respectively. From the Livingston station, 0.13, 0.12 and 0.13 was obtained as the same values respectively. Scenario 2 was proposed to tackle whether location has any impact on glitch classification or not.

IV. ANALYSIS OF RESULTS

Location	Precision	Recall	F-measure
Hanford	0.86	0.87	0.86
Livingston	0.13	0.12	0.13

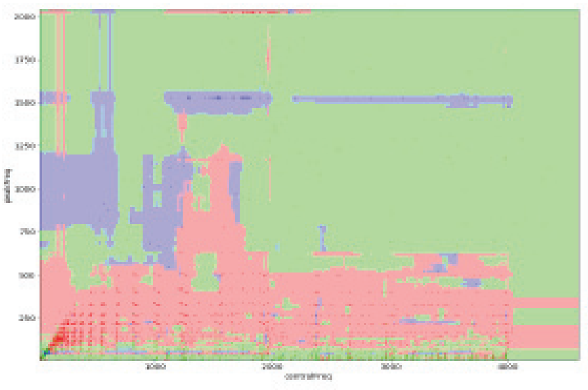


Fig. 4. Listed Color Map for Hanford data

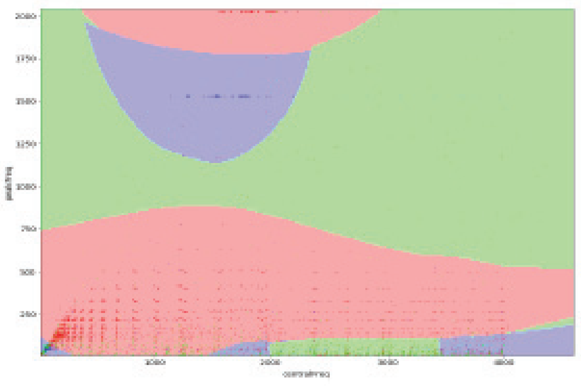


Fig. 5. Listed Color Map for Livingston data

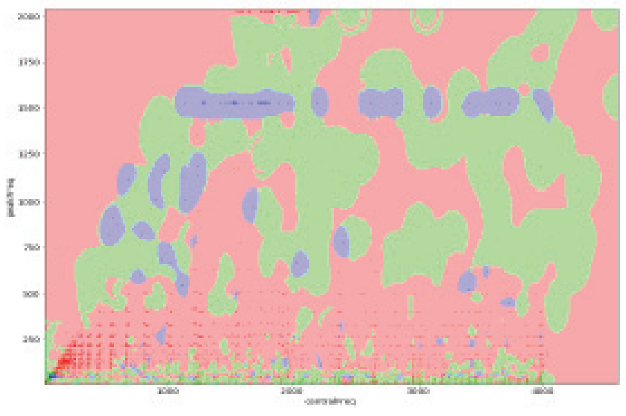


Fig. 6. Listed Color Map for Scenario 1 without location

The research problem in Scenario 2, which aimed to find out a correlation between the location of the interferometer and glitch classification hasn't been made succinctly clear through the results obtained owing to a multitude of factors that could be at play. The results hint at the accuracy in classifying data obtained from the Hanford station to be more reliable than that obtained from Livingston. This could be partially attributed to the location of the detectors since the LIGO Livingston Observatory is situated at (30°33′46.42″N 90°46′27.27″W) in Livingston, Louisiana, and the LIGO Hanford Observatory, on the DOE Hanford Site (46°27′18.52″N 119°24′27.56″W), located near Richland, Washington. The results may additionally hint at instrumental or hardware related issues in sensing or optics of light curves that are at play while collecting detector data being more prominent in the Livingston station as compared to Hanford. The Listed Color Maps show the variations in visualizing the data distribution after running the model with the highest f-score. Over-fitting of data, and most importantly imbalance might also be a contributing factor.

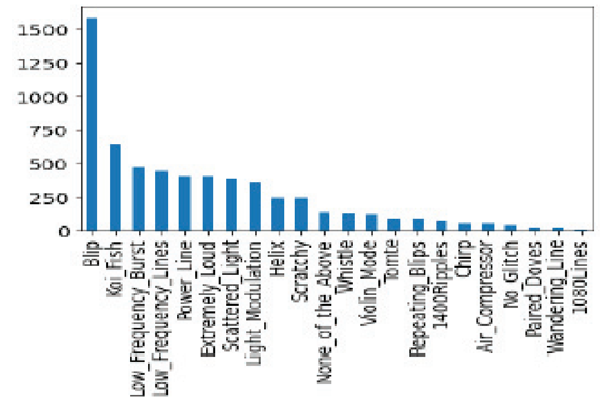


Fig. 7. Distribution of Glitches, their types

V. Discussions and Conclusion

As can be seen from the Glitch distribution bar plot, the number of non glitch events is extremely low and the percentage of Blip glitches is very high. The very innate nature of imbalanced data obtained from the interferometers at LIGO prompts us to consider up-sampling and other data balancing techniques which haven't been considered in this project. The observation that one cannot seem to obtain a highly robust glitch classification model hints at how difficult it is in terms of the laws of physics and nature to identify black hole mergers which cause gravitational waves. The very minute proportion of 'No glitch' labels shows the rarity in detection, and hence re-sampling techniques may not help us infer much knowledge about the detection events. An optimum solution for the Gravitational Wave community would be to approach this problem using a combined pipeline of adopting resilient hardware as well as robust Machine Learning networks which show promise when combined with emerging techniques like Spiking Neural Networks, Quantum Computation, etc.

Acknowledgements

The author would like to thank Dr Tanmay Basu, Assistant Professor at the Department of Data Science and Engineering, Indian Institute of Science, Education and Research, Bhopal for his guidance.

References

[1] https://ml4physicalsciences.github.io/2017/files/nipsdlps201725.pdf

[2] https : //github.com/zerafachris/g2net2ndtrainingschoolmaltamar2020/blob/master/HACKleaderboard/HACKBaselineModel/99MLRandomForestBaseline.ipynb

[3] https : //iopscience.iop.org/article/10.1088/1742 − 6596/243/1/012006/pdf

[4] https : //arxiv.org/abs/1811.03867

[5] https : //arxiv.org/abs/2207.04001

[6] https : //ml4physicalsciences.github.io/2017/files/nipsdlps201725.pdf

[7] https : //ivpl.northwestern.edu/wp − content/uploads/2019/02/1 − s2.0 − 0020025518301634 −main.pdf

[8] https : //www.g2net.eu/wp − content/uploads/2021/07/PowellMay2021.pdf

[9] http : //noiselab.ucsd.edu/ECE228 − 2020/projects/Report/89Report.pdf