Work in Classification

Simone Albanesi,[1, *] Marina Berbel,[2, †] Marco Cavaglià,[3, ‡] Lorena Magaña Zertuche,[4, §] Miquel Miravet-Tenés,[5, ¶] Dimitra Tseneklidou,[6, **] and Yanyan Zheng[3, ††]

[1]*Dipartimento di Fisica, Universit'a di Torino & INFN Sezione di Torino, via P. Giuria 1, 10125 Torino, Italy*
[2]*Departament de Matemàtiques, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain*
[3]*Institute of Multi-messenger Astrophysics and Cosmology & Physics Department, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[4]*Department of Physics and Astronomy, University of Mississippi, University, Mississippi 38677, USA*
[5]*Departament d'Astronomia i Astrofísica, Universitat de València, Dr. Moliner 50, 46100, Burjassot (València), Spain*
[6]*Theoretical Astrophysics Department, Eberhard-Karls University of Tübingen, Tübingen 72076, Germany*

(Dated: January 18, 2023)

Abstract goes here.

I. INTRODUCTION

In our introduction we want to start from big picture (LIGO, ML, pipelines) and narrow down to what work we are presenting here and why it is important. We also describe in which ways it is novel and how it compares to previous works like in [**?**]. We may also want to cite [**?**].

The breakthrough observation of the binary neutron star (BNS) merger simultaneously in gravitational and electromagnetic (E/M) waves [**?**] marked a new era of multi-messenger astronomy including gravitational waves (GWs). The event GW170817 gave answers to open questions, such as the origin and production of Gamma-ray bursts (*cite), the production of heavy elements during a merger of BNS (*cite), rule out Equations of States (*cite) and alternative theories of gravity (*cite), +?. Among the challenges that it brings is the early alert of the E/M telescopes. The sources that can produce an E/M counterpart are the coalescences of a neutron star with another neutron star (NSNS) or with a black hole (NSBH). In the case of NSNS: GRBs, neutron star. For NSBH: accretion disk –> short GRBs.

- Mention empirical fits, Foucart?
- Real time inference: What was done in O2, O3 (Deep)
- Probabilities HasNS, HasRemnant
- What we suggest, Random Forest classification (what is it, what are the advantages)

II. DATASET AND LABELING

HasNS true if m2inj<3Ms

HasREM Focault remnant mass >0. Requires compactness so it is model dependent. The O2 dataset used 2H EoS. Later we do the training with 23 EoS and do a weighted average according to the Bayes factor of each EoS, obtaining analogous performance.

Until here is same as Deep.

NEW: Instead of training two different classifiers, one for each feature, as they are in part related (an event cannot have remnant if there is no NS) we build 3 mutually exclusive categories: label 0 if no NS and no remnant, label 1 if NS but no remnant, label 2 if both. This labeling eliminates the posibility of an unphysical classification of the event, where pHasRem>pHasNS, as there is no category for hasREM but no NS. As the categories are mutually exclusive $p(0) + p(1) + p(2) = 1$. Trivially p(HasREM)=p(2). And for the NS p(hasNS)=p(hasNS and hasREM ∪ hasNS and No hasREM)=p(2∪1)=p(2)+p(1)=1-p(0)

HasNS	HasRem	Our label
0	0	0
1	0	1
1	1	2

TABLE I. Labelling adopted for classification of having a NS and having a remnant with the same classifier

We test the performance using random forest (RF) and compare the results with k-nearest neighbors (KNN) [paper Deep] and genetic programming (GP) [paper GP].

135538 training samples. 58088 for testing. Training/testing dataset: injections in O2 data stream. (Injections time and waveform approximants detailed in Chatterjee et al 2020). 23 eos, change the rem label

III. APPROACHING CLASSIFICATION THROUGH ML

KNN was used. We try more algo

* simone.albanesi@edu.unito.it
† mberbel@mat.uab.cat
‡ cavagliam@mst.edu
§ lmaganaz@go.olemiss.edu
¶ miquel.miravet@uv.es
** dimitra.tseneklidou@uni-tuebingen.de
†† zytfc@umsystem.edu

A. K-Nearest Neighbors (KNN)

One of the non-parametric algorithms we have used for classification is the `KNeighborsClassifier` (KNN). This algorithm assumes that similar things are near to each other. It captures the idea of similarity by computing the distance between points in a graph. It is also a lazy learning algorithm, because it doesn't have a training phase, it uses all the data for training while classification -the dataset is stored and at the time of classification, it acts on the dataset.

The workflow of the KNN algorithm is the following: first, after splitting the data into training and testing sets, we need to select the number of neighbors, K, which can be any integer. Then, for each point of the testing dataset we compute the distance between the point and each row of the training data with a chosen metric (Euclidean, Manhattan, Chebysev, Mahalanobis), and we sort the points in ascending order based on the distance value. By choosing the top K neighbors from the sorted array, we assign a class to the test point, which is the most frequent class of the chosen neighbors.

IV. RANDOM FOREST (RF)

A RF is an ensemble of decision trees. One of its major strengths is that every train trains and classifies independently from the rest, while the RF classification joints all the results and assign as category the mode from the trees. The probability of belonging to a category is therefore straightforward, being the number of trees that chose it divided by the total number of trees. Notice that the training and evaluation of a RF can be accelerated by parallelization, as computations inside each tree are independent from the rest.

The training of a RF is usually done with bootstrap, a technique that assigns a random subset of the training dataset to each tree. This prevents overfitting as every individual classifier is not exposed to the same data, and encourages pattern recognition by studying the same data from different subsets. Every decision tree is composed by nodes, where data its splitted until the different categories are separated. At each node, a subset of the features of the data is selected along threshold values that maximize the information gain at the separation. The binary splitting at each node gives the tree its name, as it can be visualized as roots going deeper at separations.

We use the RK implementation in scikitlearn [REF AND DETAILS]. The main hyperparameters to tune in this module are the number of trees, the maximum depth allowed and the information gain criteria used at splitting (two are offered). We have observed that the maximum number of features to be considered in a node can be kept fix as the square root of the total number of features. Given that the aiming of this work is to improve the current low latency classification, the model once trained can occupy a restricted amount of memory. Therefore before searching the optimum hyperparameters for our dataset, we restrict those which make heavier models: the number of trees and their depth. We set to 100 the maximum number of trees the forest may have, and 25 their possible maximum depth.

	Best					S	
EOS	**Trees**	**Depth**	**Score**	**MB**	**MB (c)**	**Trees**	**Depth**
APR4_BB	300	15	0.9683018	94.7	19.7	50	15
BHF_BBB2	80	15	0.9685127	24.4	5.1	300	15
H4	80	15	0.9618587	29.6	6.1	300	15
HQC18	300	15	0.9673755	93.7	19.6	100	15
KDE0V	300	15	0.9673295	92.0	19.3	80	15
KDE0V1	100	15	0.96704954	30.9	6.5	80	15
MPA1	80	15	0.96601225	27.2	5.6	300	15
MS1_PP	300	15	0.96563534	113.5	23.2	80	15
MS1B_PP	300	15	0.96555340	114.2	23.3	100	15
RS	300	15	0.96447350	103.8	21.6	80	15
SK255	300	15	0.96472405	105.8	22.0	100	15

TABLE II. table1 comparison forests

For the RF we use events with 5 features: the two masses, their corresponding spins and the SNR of the detection. In the tuning of the hyperparameters we measure the performance by its score: the number of events correctly classified against the total number of events in the testing dataset, if threshold is taken as 0.5. As all categories are balanced, this approach is enough to roughly compare models. The best model found achieves a score of *whatever* using *number* of trees, *number* maximum depth and *name* criteria for the information gain.

INFORMATION TO SAY, JUST HERE AND NOT WELL WRITTEN:

For the 23 EoS we try a crossvalidation with:

trees in the forest = [10, 30, 50, 80, 100, 300] (more trees take too much memory) criterion = 'entropy' (in all tests, criterion 'gini' gives very similar results)

max_features = 'sqrt' (in all tests, using a portion of the features instead of all of them in each node gives better results)

max_depth = [15, 25, 35, 45, None] (we would prefer more shallow trees)

Using the complete dataset, and not Sushant partitions, Use first 30

We obtain the following:

And therefore we select 50 trees, 15 depth for all EoS and present the results with that.

	Best					Second Best					
EOS	**Trees**	**Depth**	**Score**	**MB**	**MB (c)**	**Trees**	**Depth**	**Score**	**MB**	**MB (c)**	**Δ score**
SK272	300	15	0.96401816	109.0	22.6	100	15	0.96381927	36.4	7.5	1.99e-4
SKI2	50	15	0.96242338	18.8	3.9	300	15	0.96233966	112.8	23.2	8.37e-5
SKI3	50	15	0.96174537	19.0	3.9	100	15	0.96167916	38.1	7.8	6.62e-5
SKI4	300	15	0.96598969	100.6	20.9	30	15	0.96590604	9.8	2.1	8.37e-5
SKI5	100	15	0.96343381	38.2	7.8	80	15	0.96331793	30.4	6.2	1.16e-4
SKI6	300	15	0.96586928	101.7	21.1	30	15	0.96565242	10.0	2.1	2.17e-4
SKMP	300	15	0.96544567	100.2	20.9	80	15	0.96527695	26.9	5.6	1.69e-4
SKOP	100	15	0.96610459	32.3	6.8	300	15	0.96603605	96.2	20.1	6.85e-5
SLy	80	15	0.96728884	25.3	5.3	300	15	0.96720392	95.2	19.9	8.49e-5
SLY2	100	15	0.96745868	31.8	6.6	80	15	0.96722091	25.4	5.3	2.38e-4
SLY9	300	15	0.96605993	101.6	21.1	100	15	0.96590909	34.1	7.1	1.51e-4
SLY230A	300	15	0.96714915	95.5	20.0	100	15	0.96689580	31.9	6.7	2.53e-4

TABLE III. table2 RF comparison

EOS	Score	$N_{\text{neighbors}}$	Bayes Factor
APR4 EPP	0.953	10	1.526
BHF BBB2	0.951	8	1.555
H4	0.947	10	0.056
HQC18	0.950	10	1.422
KDE0V	0.951	8	1.177
KDE0V1	0.949	10	1.283
MPA1	0.951	14	0.276
MS1 PP	0.948	12	0.001
MS1B PP	0.947	11	0.009
RS	0.945	10	0.176
SK255	0.946	10	0.179
SK272	0.945	10	0.156
SKI2	0.943	12	0.108
SKI3	0.943	12	0.107
SKI4	0.949	10	0.330
SKI5	0.945	9	0.025
SKI6	0.949	10	0.288
SKMP	0.948	10	0.290
SKOP	0.948	10	0.618
SLy	0.950	10	1.000
SLY2	0.950	10	1.028
SLY9	0.949	10	0.370
SLY230A	0.950	10	0.932

TABLE IV.

V. RESULTS

A. KNN Results

We apply cross validation in order to fix the different hyperparameters of the algorithm. To do so, we compute the score over a range of different parameters. We consider a number of neighbors between 1 and 20; the different metrics we test are the *euclidean*, *manhattan* and *cityblock*; the algorithms to compute the nearest neighbors can be `BallTree`, `KDTree` and the brute-force search, and the possible weight functions are the uniform weights (all points are weighted equally) and the *distance* weights (points are weighted by the inverse of their distance).

Considering all these possibilities, we apply the `cross val score` function from `scikit-learn` using a 10-fold cross validation to all the 23 datasets with different EOS. We find that the optimal metric, algorithm and weights are the same for all the EOS, being the *manhattan* metric, the `BallTree` algorithm (with a leaf size of 30) and the *distance* weights. The only parameter that differs is the number of neighbors, that goes from 8 to 12. One can find the optimal hyperparameters and the corresponding score for each EOS in Table ??. Doing an average weighting by the Bayes factor of each EOS, we finally get an optimal number of neighbors of 10. The mean score from the cross validation goes from 0.941 (for H4 EOS) to 0.953 (for APR4 EPP), as one can see in Table ??.

Considering now the SLy EOS, the model with these parameters gives a confusion matrix that is shown in Fig. ??. The probability of having a NS as a function of m_1 and m_2 is shown in Fig. ??. There are no big differences with different values of the spins, but the most remarkable one is that the model classifies better for 0-spin values, especially when m_1 is large. There is also a dependence of the probability of having a remnant on the value of the spin that can be seen in Fig. ??.This dependence is correct, since the probability of having a remnant increases for large values of m_1 at larger spin, but this dependence is given by the EOS.

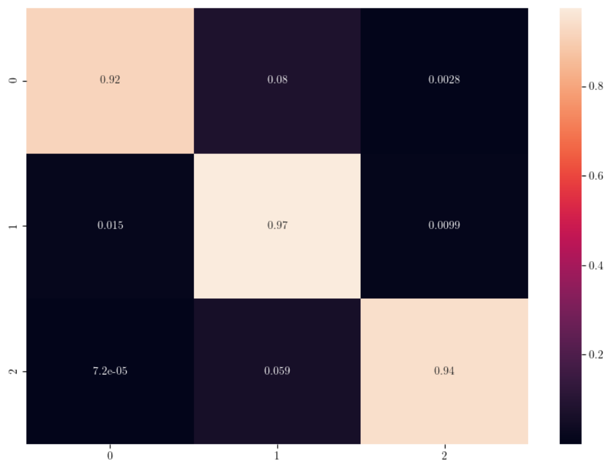

FIG. 1. Confusion matrix SLy KNN

In Figs. ??, ?? and ?? we depict the histograms of the probabilities p(*HasNS*) (p(*HasRemnant*)) for injections of binaries that had a NS (EM counterpart) and for those that not, for the EOS BHF BB2, MS1 PP and SLy, respectively.

We show in Fig. ?? the ROC curves of the classifier for the two different probabilities we consider. All the EOS are depicted in this figure, but we highlight three cases: BHF BBB2, MS1 PP and SLy.

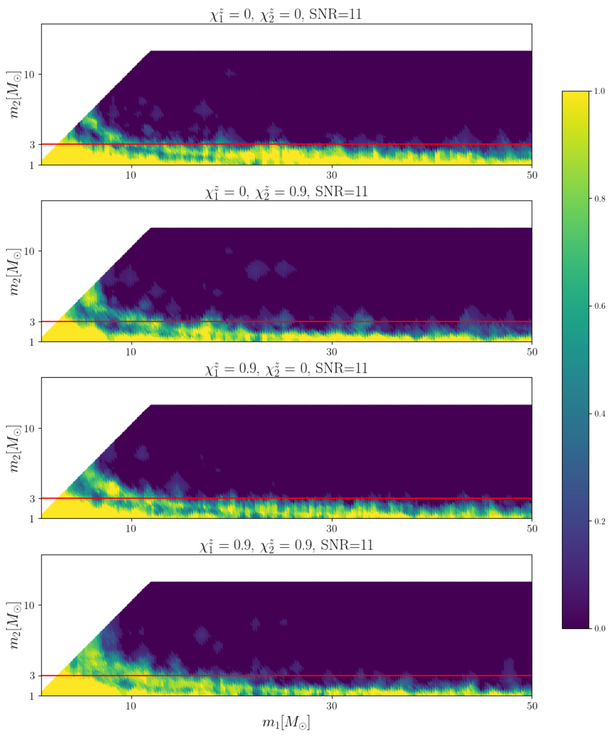

FIG. 2. Parameter sweep NS SLy KNN

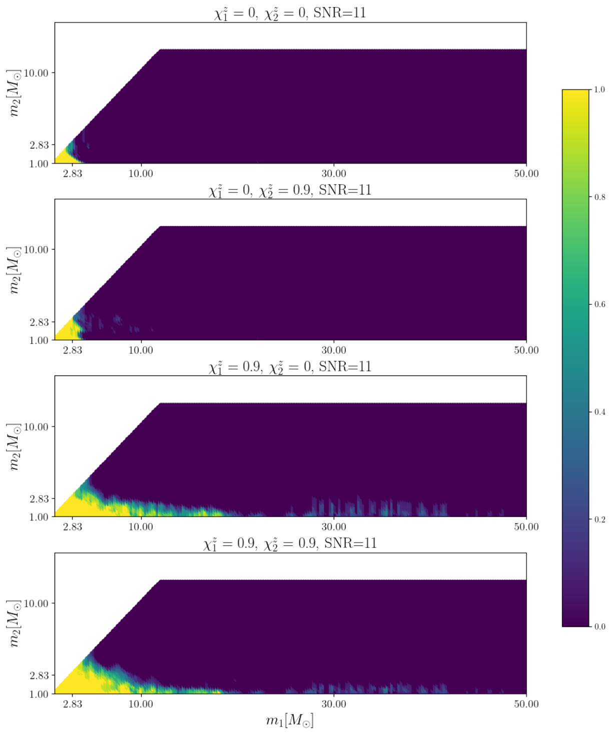

FIG. 3. Parameter sweep REM SLy KNN

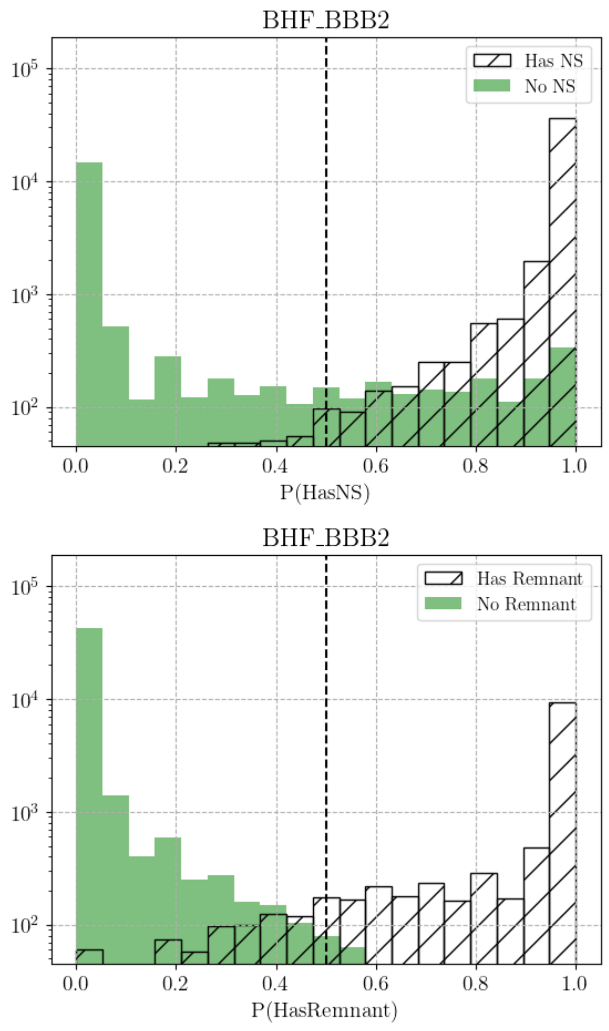

FIG. 4. Histograms BHF BBB2 KNN

	Has NS				Has REM			
	RF		KNN		RF		KNN	
Threshold	TP	FP	TP	FP	TP	FP	TP	FP
0.1	0.999	0.107	0.999	0.156	0.998	0.011	0.992	0.051
0.3	0.998	0.068	0.996	0.117	0.993	0.005	0.974	0.017
0.5	0.994	0.042	0.991	0.088	0.985	0.003	0.937	0.006
0.8	0.967	0.014	0.966	0.043	0.957	0.001	0.845	0.001

TABLE V. BHF_BB2

B. RF Results

VI. CONCLUSIONS

Reiterate why what we did is important and how it improves current knowledge.

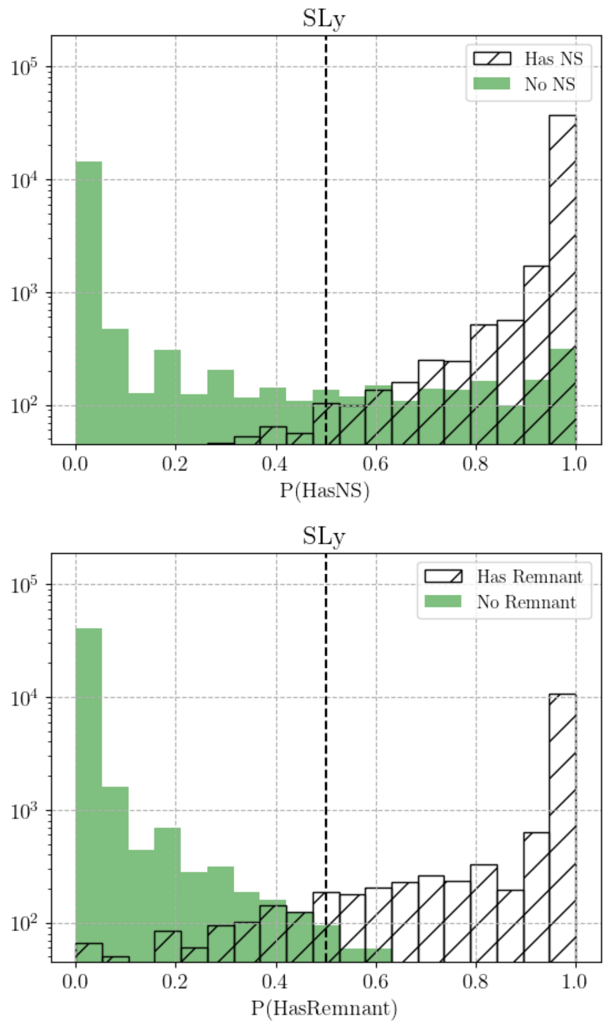

FIG. 5. Histograms SLy KNN

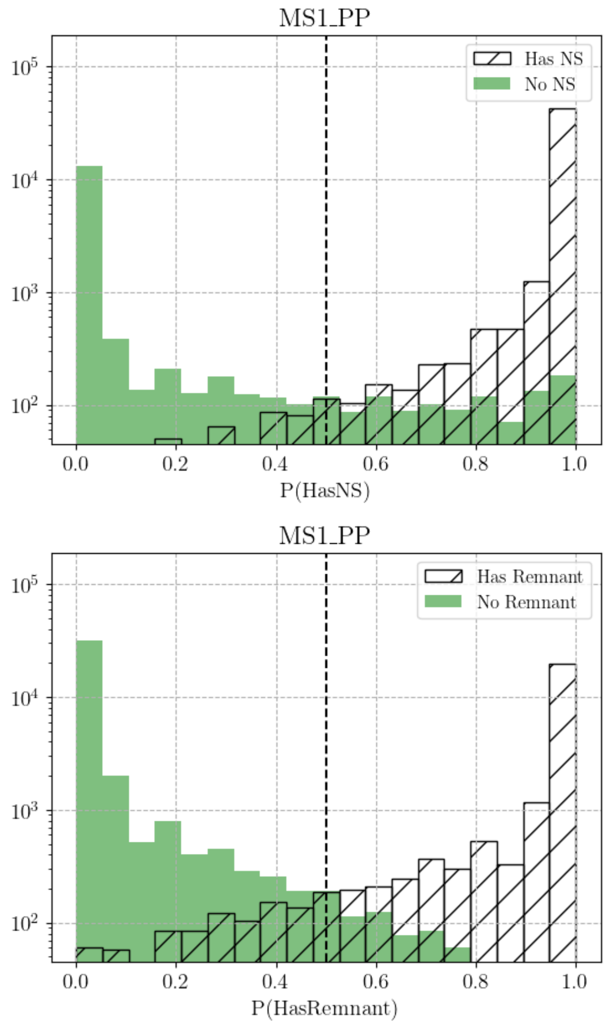

FIG. 6. Histograms MS1 PP KNN

	Has NS				Has REM			
	RF		KNN		RF		KNN	
Threshold	TP	FP	TP	FP	TP	FP	TP	FP
0.1	1.000	0.114	0.999	0.138	0.999	0.023	0.995	0.103
0.3	0.998	0.065	0.995	0.097	0.996	0.010	0.983	0.044
0.5	0.994	0.036	0.989	0.068	0.990	0.004	0.961	0.019
0.8	0.968	0.011	0.967	0.031	0.967	0.001	0.899	0.004

TABLE VI. MS1_PP

	Has NS				Has REM			
	RF		KNN		RF		KNN	
Threshold	TP	FP	TP	FP	TP	FP	TP	FP
0.1	1.000	0.107	0.999	0.155	0.998	0.013	0.992	0.059
0.3	0.999	0.064	0.996	0.112	0.993	0.005	0.974	0.020
0.5	0.994	0.038	0.990	0.084	0.986	0.003	0.940	0.007
0.8	0.965	0.012	0.965	0.040	0.958	0.001	0.848	0.001

TABLE VII. SLy

ACKNOWLEDGMENTS

We thank Deep Chatterjee for useful discussions and for sharing his work, which helped us compare our results to those of [**?**]. We also thank Shaon Ghosh for useful discussions and X, Y, and Z for reviewing an earlier version of this manuscript. Part of this research was performed while the authors were visiting the Institute of Pure and Applied Mathematics (IPAM), University of California Los-Angeles (UCLA). The authors would like to thank IPAM, UCLA and the National Science Foundation through grant DMS-1925919 for their warm hospitality during the fall of 2021.

The authors are grateful for computational resources provided by the LIGO Laboratory and supported by the U.S. National Science Foundation Grants PHY-0757058 and PHY-0823459, as well as resources from the Gravitational Wave Open Science Center, a service of the LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration.

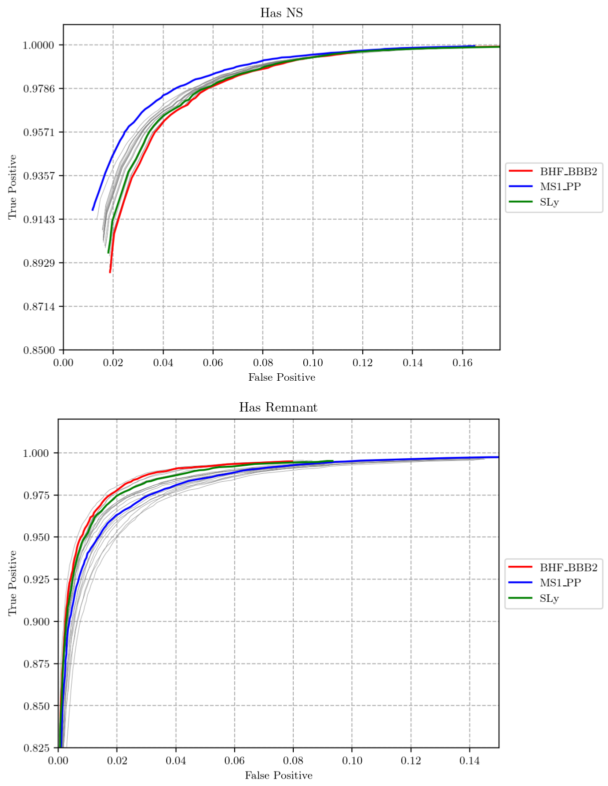

FIG. 7. ROC curves KNN

The work of L.M.Z. was partially supported by the MSSGC Graduate Research Fellowship, awarded through the NASA Cooperative Agreement 80NSSC20M0101. The work of X.Y. was partially supported by NSF Grant No. PHY-20XXXXX. The work of M.M.T. was supported by the Spanish Ministry of Universities through the Ph.D. grant No. FPU19/01750, by the Spanish Agencia Estatal de Investigación (Grants No. PGC2018-095984-B-I00 and PID2021-125485NB-C21) and by the Generalitat Valenciana (Grant No. PROMETEO/2019/071)

The work of M.B. was supported by the Spanish Agencia Estatal de Investigación (Grants No. PID2020-118236GB-I00).

All plots were made using the python package `matplotlib` [?].

This manuscript has been assigned LIGO Document Control Center number LIGO-P22XXXXX.

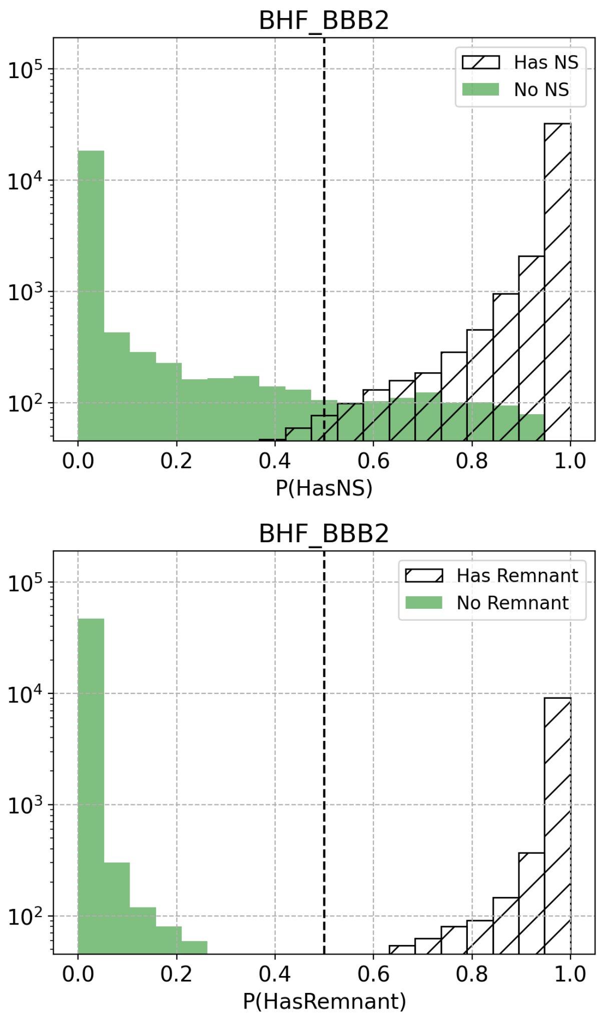

FIG. 8. Histograms BHF BBB2

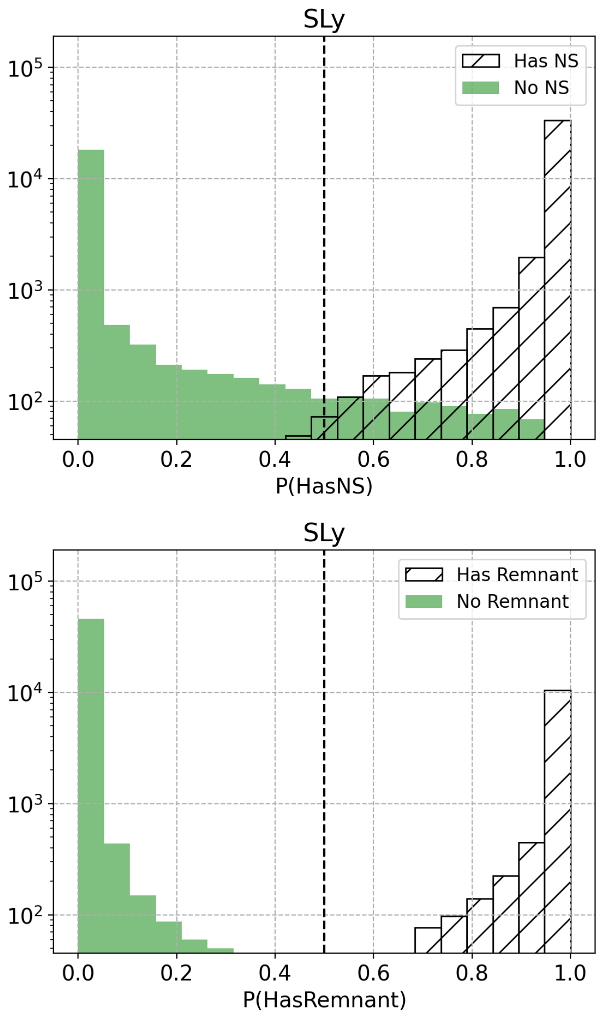

FIG. 9. Histograms SLy

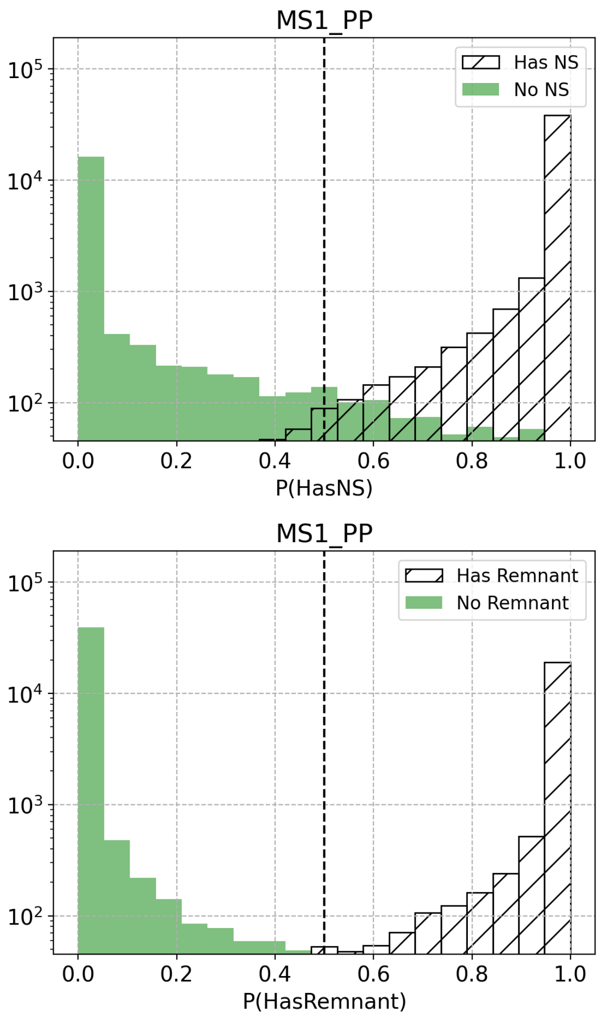

FIG. 10. Histograms MS1 PP

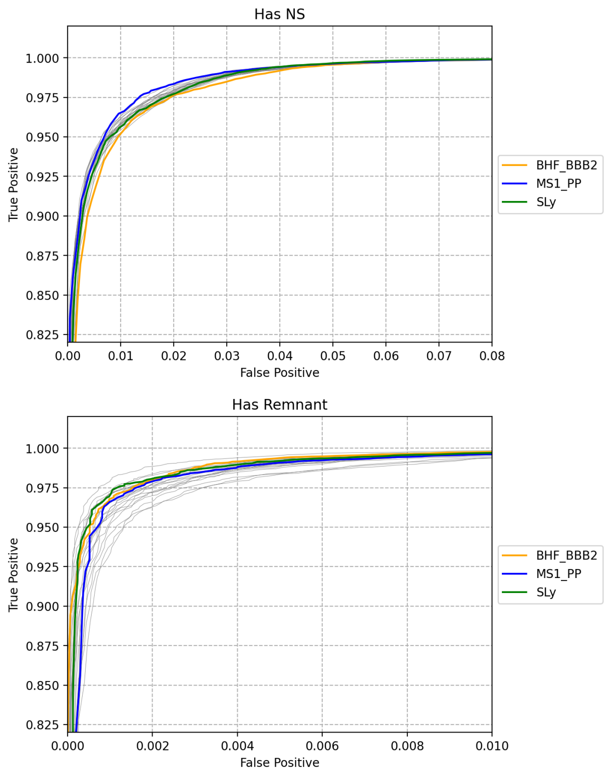

FIG. 11. ROC curves