IL NUOVO CIMENTO **48 C** (2025) 99
DOI 10.1393/ncc/i2025-25099-8

Colloquia: IFAE 2024

Machine learning techniques for gravitational waves data analysis(*)

L. Mobilia(1)(2)(**) and the MBTA Collaboration

(1) *Sez. di Fisica, Università degli Studi di Urbino Carlo Bo - Urbino, Italy*
(2) *INFN, Sezione di Firenze - Firenze, Italy*

received 2 December 2024

Summary. — The use of machine learning in the study of gravitational wave physics is increasingly widespread. The flexibility and results that this technology has achieved encourage the use and exploration of such techniques in this research field. In this work, we develop a machine learning tool based on the random forest technique to enhance the measurement capabilities of the MBTA (Multi-Band Template Analysis) algorithm in distinguishing signal from noise. The results are obtained by considering different configurations and features, taking into account both physical and statistical values of the triggers to train and test the machine learning algorithm. Comparisons between the statistical significance obtained from machine learning and the classical algorithm were conducted using real data.

1. – Introduction

The first detection of a gravitational wave signal [3] obtained by the LIGO interferometers opened the door to the gravitational astronomy [1]. Since then, more than 90 events have been measured [6], [7], [8], [9] by the LIGO-Virgo-Kagra collaboration. After the discovery of the first binary neutron star system [4], the multi-messenger gravitational wave astronomy has become a reality, with important consequences for cosmological studies and new physics [2], [5]. Those results have been achieved through the interferometers LIGO [10] in the United States of America and Virgo [11] in Italy. In order to detect the gravitational waves signals several algorithms have been developed, for both modelled and un-modelled search [15]. For the compact binary coalescence (CBC) online analysis, that consists in the search of astrophysical compact objects such as binary black holes, a particular type of pipelines that relies on the matched-filtering method are required [12], [13], [14]. In this contribution, we will try to increase the detection capability of the pipeline MBTA, a CBC pipeline currently used for the online analysis, by adopting machine learning techniques.

(*) IFAE 2024 - "Cosmology and Astroparticles" session
(**) E-mail: l.mobilia@campus.uniurb.it

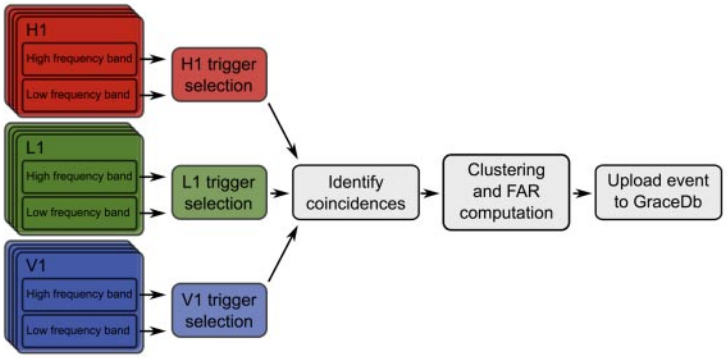

Fig. 1. – Scheme of the MBTA pipeline: matched-filtering for each band is evaluated from the detectors' stream then, if a trigger occurs, the false alarm rate is evaluated and the event is uploaded to the database.

2. – MBTA

The Multi-Band Template Analysis (MBTA) pipeline is an algorithm based on the matched-filtering technique for the compact binary coalescence events detection [16], [17]. In order to work, the matched-filtering methods requires a 'bank' of templates that is built with an hybrid-code [18]. MBTA analyses the data stream provided by the output of the interferometers considering chunks of several seconds each. In each chunk the matched-filtering is applied. In MBTA, this procedure is done independently and in parallel for both low and high frequencies, dividing accordingly the template bank and so reducing considerably the computational time. If the matched-filtering analysis results in a signal to noise ratio value above a certain threshold for both the frequency bands, then a triggers is produced. Each trigger has several parameters' such as the signal to noise ratio ρ, the χ^2 or the masses and the spins. The matched-filtering technique is optimal in case of Gaussian noise, but for real noise data, the presence of glitches must be taken into account. In the actual configuration, MBTA in order to distinguish glitches artefacts from effective gravitational wave signals, considers some of the triggers' parameters to build a ranking statistic named ρ_{rw} that increases the separability between noise and signal. To ensure robust statistics for potential astrophysical events, synthetic gravitational wave signals (injections) are added to the data stream, allowing the differentiation between glitches and injections.

3. – Machine learning for gravitational waves - random forest algorithm

The use of machine learning is becoming more and more popular in the field of gravitational waves [19]. In particular, algorithms for supervised learning such as random forest [20] and neural network [21] have been explored in the context of CBC analysis [22]. In particular in the work [23] a random forest algorithm has been trained using several features and compared the statistical significance obtained by the pipeline with respect to the one resulting from the machine learning. A random forest algorithm is a collection of binary classifiers (trees) that divide the features' space in rectangles and fit a simple model in each one. The idea is to use many of these classifiers in order to obtain a powerful final committee. The randomness is introduced in order to reduce the variance,

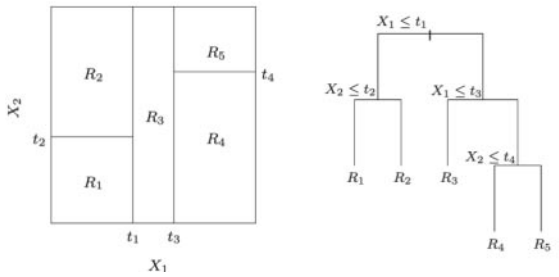

Fig. 2. – The tree tries to divide the features (X_1, X_2) space fitting a simple model in order to correctly classify an input.

and it is obtained by randomizing the data input for each classifier through bagging procedure. To build its statistical significance, each tree is trained on a set of labelled input data, each with features $\vec{x}$. Afterwards, the model is verified over a test dataset, containing unlabelled data, and it assigns to each datum a score $p_s \in [0, 1]$, where values near 0 or 1 represents the confidence the machine has to assign an event to one of the two classes. Formally, given a single tree T, it will give a score $\hat{p}_s$ according to the feature $\vec{x}$, so given an ensemble of N trees $\{T_i\}_{i=1}^{N}$, then the final score will be provided just by the averages of the scores of each tree $p_s = \frac{1}{N}\sum_{i=1}^{N} \hat{p}_{s,i}(\vec{x})$.

4. – O3a dataset

In this work have considered the triggers from the O3a data acquisition campaign, from April 1st 2019 to October 1st 2019. We considered the HL coincidence triggers, that are the triggers provided by MBTA when the H and L detectors are locked and acquiring data. To be claimed as a trigger, the signal to noise ratio of single triggers must be $\rho_H, \rho_L > \rho_{min}$, where ρ_{min} is a threshold value and the time between the triggers is in the coincidence window (less than $15ms$ between triggers). For what concerns the injections, a triggers is claimed to be associated to a synthetic event if there are triggers around $[-80, 40]ms$ the time of the injection. Among these, the loudest one is chosen as injection trigger. The dataset consists in 83532 injections and 43923 noise events. For training and test, we split respectively the dataset in 70% for training and 30% for testing, resulting in

5. – Features and hyper-parameters

The package used for this work is `scikit-learn` [24]. As hyper-parameters we define the set of parameters that characterize the machine learning algorithm. The choice of those is fundamental in order to have an efficient classifier. This selection has been made

TABLE I. – *Training and test dataset.*

	Training	Test
Noise	30643	13280
Injections	58576	24956

TABLE II. – *Hyper-parameters and features.*

Number of trees	3000
Impurity criterion	'gini'
Maximum number of features	'sqrt'
Maximum depth of each tree	24
Minumum samples leaf	12
Physical features	m_1, m_2, s_{1z}, s_{2z}, t_d
Statistical features	L_{snr}, H_{snr}, χ^2_L, χ^2_H, ER_L, ER_H, L_ϕ, H_ϕ, L_d, H_d

considering several possible combinations of those and selecting the one that maximizes the receiver operating characteristic curve. Also, the triggers' features have been picked up following the same principle. This work results in the choice of hyper-parameters's set and features listed in table II.

The hyper-parameters chosen involve both the structure of the single tree and the forest. For the single tree in fact, the impurity criterion defines if and how to split the node in two sub-leaves. The maximum number of features refers to the number of features on which each tree will be trained on. The maximum depth is related to the nodes that each tree can build before get stop and similarly the minimum samples leaf is the minimum number of samples that must fall in the child nodes in order to create that. All these values have been chosen empirically in order to avoid over-fitting, that may happen if the model is free to grow indefinitely. The features have been picked considering both statistical and physical parameters of the triggers. The masses and the spins have been inserted since they will provide information about which templates may be 'noisy' during the matched-filtering procedure, also since we expect the glitches to be fast, the time duration of the trigger has been introduced. The statistical features contain both the significance of the triggers, through the signal to noise ratios of both the interferometers and the χ^2 values, and information about the noise at the moment of the detection, via the $ER_{L,H}$ parameters. Those latter parameters are defined as the excess at the rate of triggers compared to the rate of surviving triggers once the χ^2 cut is applied. Finally, the phase ϕ and the distance d measured by both interferometers are considering to see if they contribute in distinguishing noise from signals.

6. – Results

Once trained the machine using the configuration and the features specified in the previous section, the model is tested on the test dataset in order to study its performances. In order to compare the random forest output with respect to the detection statistics provided by MBTA, the false alarm probability α_s and the number of detections N_d have been taken into account. Those are defined as

$$\alpha_s = \frac{1}{N_n} \sum_{i=1}^{N_n} \theta(p_s^i - \bar{p}_s) \tag{1}$$

$$N_d = \sum_{i=1}^{N_s} \theta(p_s^i - \bar{p}_s) \tag{2}$$

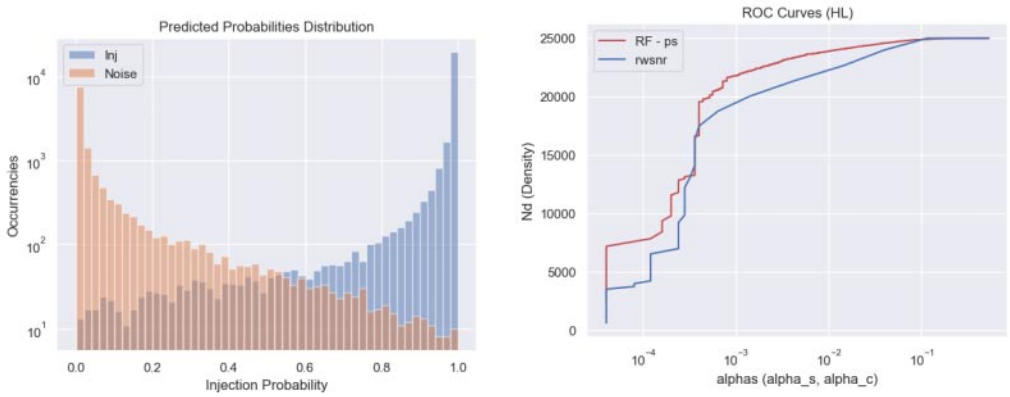

Fig. 3. – Noise (a) and injection (b) recovered by MBTA during O3a data acquisition campaign. The statistical significance obtained by MBTA is reported, and the gps times of each trigger are reported respectively on the y and x axis.

The former (α_s) is the fraction of numbers of noise events that passes the selection with a statistical significance p_s greater than the given threshold $\bar{p}_s$. This is computed using the Heaviside function θ. The latter (N_d), is the number of detections (injections) with a statistic p_s greater than the threshold $\bar{p}_s$. The same values can be obtained from the MBTA statistics considering instead of the p_s, the amplitude ρ_{rw} that is a function of the statistical significance of both detectors H and L and of the χ^2 values.

We see (fig. 3) that the curve provided by the algorithm is above the one obtained by the pipeline, this claims that at a given value of false alarm probability, the machine obtains a greater value of injected signals with respect to MBTA. This also results in an increasing the distance achievable by the pipeline (fig. 4). If we consider for instance a cut at $\alpha_s = 10^{-3}$ we obtain 21419 injections with respect to the 18063 resulting from the pipeline.

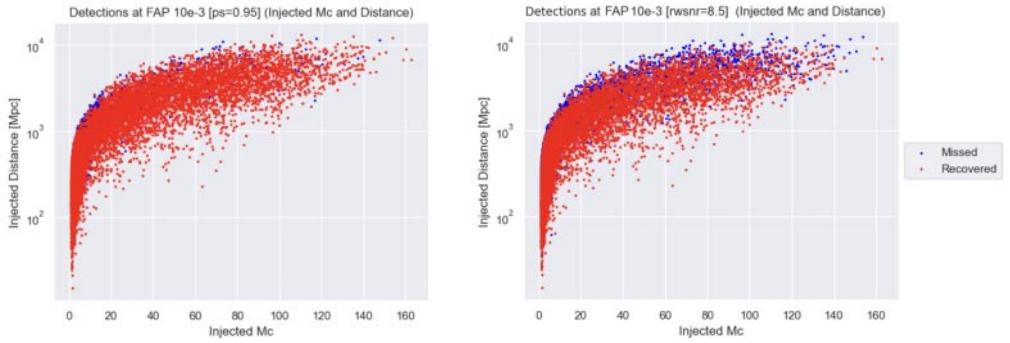

Fig. 4. – Distance obtainable at fixed false alarm probability α_s by MBTA's statistical significance (a) and by machine learning (b).

7. – Conclusions

In this work we, analysed the capability of machine learning in boosting the MBTA's pipeline detection capabilities using a random forest algorithm. The results seem to be encouraging in exploring this new tool for the CBC detection's pipeline MBTA. The machine obtains a detection statistic that is compatible or superior with respect to MBTA. This results in increasing the detectable distances, as stated by fig. 3. In any cases, further studies to increase the interpretability of the detection statistics must be performed. In order to build a tool that is effectively useful in the detection, we must be sure how the features and the hyper-parameters impact the model, also avoiding overfitting is fundamental to have a reliable machine. Also, a grid or random search in order to find the best hyper-parameters combination will be helpful for optimizing the machine. A possible perspective for the future may be to define a new ranking statistic based on this machine learning approach.

REFERENCES

[1] Cai Rong-Gen *et al.*, *Natl. Sci. Rev.*, **4** (2017) 687.
[2] Abbott B. P. *et al.*, *Astrophys. J. Lett.*, **848** (2017) L12.
[3] Abbott B. P. *et al.*, *Phys. Rev. Lett.*, **116** (2016) 061102.
[4] Abbott B. P. *et al.*, *Phys. Rev. Lett.*, **119** (2017) 161101.
[5] LIGO Scientific Collaboration, `https://www.ligo.org/science/Publication-GW170817Hubble/flyer.pdf`.
[6] Abbott B. P. *et al.*, *Phys. Rev. X*, **9** (2019) 031040.
[7] Abbott R. *et al.*, *Phys. Rev. X*, **11** (2021) 021053.
[8] The LIGO Scientific Collaboration and the Virgo Collaboration, *Phys. Rev. D*, **109** (2024) 022001.
[9] Abbott R. *et al.*, *Phys. Rev. X*, **13** (2023) 041039.
[10] Aasi J. *et al.*, *Class. Quantum Grav.*, **32** (2015) 074001.
[11] Acernese F. *et al.*, *Class. Quantum Grav.*, **32** (2015) 024001.
[12] Usman S. A. *et al.*, *Class. Quantum Grav.*, **33** (2016) 215004.
[13] Sachdev S. *et al.*, `https://arxiv.org/abs/1901.08580` (2019).
[14] Aubin F. *et al.*, *Class. Quantum Grav.*, **38** (2021) 095004.
[15] Drago M. *et al.*, *SoftwareX*, **14** (2021) 100678.
[16] Adams T. *et al.*, *Class. Quantum Grav.*, **33** (2016) 175012.
[17] Andres N. *et al.*, *Class. Quantum Grav.*, **39** (2022) 055002.
[18] Ajith P. *et al.*, *Phys. Rev. D*, **89** (2014) 084041.
[19] Cuoco E. *et al.*, *Mach. Learn.: Sci. Technol.*, **2** (2020) 011002.
[20] Breiman L., *Mach. Learn.*, **45** (2001) 5.
[21] Mehlig B., *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers*, 1st edition (Cambridge University Press) 2021.
[22] Kim K. *et al.*, *Phys. Rev. D*, **101** (2020) 083006.
[23] Kapadia S. J. *et al.*, *Phys. Rev. D*, **96** (2017) 104015.
[24] Pedregosa F. *et al.*, *J. Mach. Learn. Res.*, **12** (2011) 2825.