Abstract

We present a comparison of several Difference Image Analysis (DIA) techniques, in combination with Machine Learning (ML) algorithms, applied to the identification of optical transients associated with gravitational wave events. Each technique is assessed based on the scoring metrics of Precision, Recall, and their harmonic mean F1, measured on the DIA results as standalone techniques, and also in the results after the application of ML algorithms, on transient source injections over simulated and real data. This simulations cover a wide range of instrumental configurations, as well as a variety of scenarios of observation conditions, by exploring a multidimensional set of relevant parameters, allowing us to extract general conclusions related to the identification of transient astrophysical events.

The newest subtraction techniques, and particularly the methodology published in Zackay et al. (2016) are implemented in an Open Source Python package, named properimage, suitable for many other astronomical image analyses. This together with the ML libraries we describe, provides an effective transient detection software pipeline. Here we study the effects of the different ML techniques, and the relative feature importances for classification of transient candidates, and propose an optimal combined strategy. This constitutes the basic elements of pipelines that could be applied in searches of electromagnetic counterparts to GW sources.

Keywords: keyword1 – keyword2 – keyword3

1. Motivations: Synoptic era scenario.

Synoptic sky surveys are promoters of a new era of observational astronomy, where data volume is becoming a major challenge, and discoveries are happening at a rate never experienced before. Several collaborations, involved in observational astrophysics projects, are pushing towards a data-driven science paradigm, and transforming astronomy. The Large Synoptic Survey Telescope (LSST, Ivezic and for the LSST Collaboration 2008 LSST Science Collaboration et al. 2009) is going to bring this phenomenon to a higher level, where raw data disk-space consumption is going to be in the PetaByte scales by the end of the project. To face this transformation, astronomers have been involved in information technology development for several decades, bringing to existence organizations such as IVOA1 an international alliance dedicated to organizing and making available a living archive of historical astrophysical data.

In this context, a new era of observational astronomy has begun with the first direct detection of Gravitational Waves (GW, Abbott and Collaboration 2016). This historical discovery places a huge responsibility on synoptic telescopes: the search for the electromagnetic (EM) counterparts of these GW events. The Transient Optical Robotic Observatory of the South (TOROS2), is a project aimed at identifying those GW sources. During Advanced LIGO science run O1 TOROS participated in the search for an optical counterpart to the first GW detection (Díaz and Collaboration 2016), in an effort to determine the origin of its progenitor. The theoretical scenario was developed in several articles, such as Kasen et al. (2013); Barnes et al. (2016), where models predict that a GW event like GW150914 involves a merger between compact objects. This model has three possible cases, featuring binary combinations of Black Hole and Neutron Stars components. If one component is a neutron star, the merger will produce an EM emission (or Kilonova) that will last a couple of days, and will be visible at optical and near-infrared wavelengths. The search for such an elu-

---

1 Corresponding author
Email address: bruno@oac.unc.edu.ar (B. Sánchez)
2toros.utrgv.edu
sive signature is a major challenge, and in many ways can be described as a race against time. This objective was reached recently, when the event GW170817 was identified by several collaborations as the first observed Kilonova, see for example [Abbott et al., 2017a][Díaz et al., 2017] and references therein.

One difficulty involved is the need for a comparison method between images obtained at different epochs, since a fast detection of small variations in brightness, over a large region of the sky is critical to the identification of a Kilonova event. Another issue is the requirement of a wide field of view, since the instrument dedicated to the search would need to cover several hundred square degrees per night, and also reaching deep magnitude limits. If we combine these two simple conditions of wide sky coverage and temporal resolution we have that, depending on the instrument, the image comparison method will need to deal with a variable Point Spread Functions (PSF) across several square degrees and, at the same time, be able to detect the magnitude variations with high fidelity. One of the approaches to this task is to compare the detected sources, and their brightnesses, on each epoch, and to select as possible transient candidates any mismatch. Though this methodology can find difficulties when applied on crowded stellar fields, or when the transient event is buried in a galaxy, and its flux is entangled with the luminosity profile of the host, (this was particularly the case for the event of GW170817). In the former case the angular cross match of sources can have a computational cost which would introduce an undesired time overhead for a transient discovery survey with high cadence. In the latter case in order to measure the flux and position of the transient source a correct modeling of the host galaxy luminosity profile must be applied beforehand.

There are several works that tackle this problem by using an image subtraction methodology, such as Alard and Lupton (1998); Bramich (2008). This image differentiation approach avoids the discussed issues of catalog cross-matching. A different approach has been taken by Zackay et al. (2016) in a series of three papers that derive an improved treatment of astronomical images from an statistical point of view. This is more general than previous works, and derives several commonly used astronomical image analysis methods, such as source detection, from statistical principles. The authors of this method claimed that it also reduces the necessity of a posterior Machine Learning (ML) analysis. The astronomical community is using ML in new problems every year, seizing its capability for solving data processing issues based on handcrafted examples (Fortson et al., 2012), especially in the transient detection area (Djorgovski et al., 2010; Law et al., 2009; Rau et al., 2009).

In this work we implement three difference image analysis techniques: Alard & Lupton’s (from now on A), Bramich’s (B), and Zackay’s methodology in two separated ways (Z and S) to develop and establish the best possible combination of difference imaging and machine learning techniques based on a comprehensive metric. Novel promising methods such as those based on Convolutional Neural Networks (Bai, 2017) will be compared in the future, when the TOROS collaboration have produced enough data to train such complex models. In this small survey context classical Machine Learning algorithms should to be suitable enough.

In the following section we introduce the difference image and Machine Learning techniques to be studied, in section 3 we present the simulated and real data sets that we will be used in the analysis. In section 4 we discuss the results of difference image analysis techniques prior to the implementation of Machine Learning algorithms. After that we perform the feature selection, and analyze different ML algorithms, estimate their performance for the classification of real/bogus detected on difference imaging results. Finally in section 5 we present concluding remarks and future prospects over this work. The software developed and datasets here used are open source, and can be found on TOROS public repositories.

2. Methods

2.1. Difference Image Analysis

Difference Image Analysis (DIA) is a technique which directly compares two images of the same position in the sky, taken at different epochs. It is usual that one of these is a co-addition of many previously taken images, and has very high signal to noise ratio, known as reference frame (R). The other image would be the recently acquired new image (N). Both images are assumed to be astrometrically aligned and registered, and so a special kind of subtraction is performed to deliver, after object detection, the transient candidates. The detection procedure in the difference images is in essence a classification problem (source or background), that gives as a result detections of transient sources (TS), detection of artifacts (Ar) and missed transients (MT).

2.1.1. Linearized kernel models

Image differencing goes back to Phillips and Davis (1995) where the Eq. 1 linking the new and the reference images at position (x, y), is proposed by means of a deconvolution kernel (\(\text{Ker}(a, v)\)), bound directly to the change in shape of the PSF between both images. A direct solution would be of the form of Eq. 2 by solving in Fourier space for the kernel (here \(\hat{A}\) denotes the Fourier transform of A).

\[
R(x, y) \otimes \text{Ker}(a, v) = N(x, y) \quad (1)
\]

\[
\hat{\text{Ker}} = \frac{\hat{N}}{\hat{R}} \quad (2)
\]

This can become numerically unstable in the case that the PSF of \(N\) is narrower than the PSF of \(R\) image, and also in the presence of high frequency noise on \(R\), making essential to the success of this methodology the good quality of the Reference images. The linearized kernel model was extensively developed in Alard and Lupton (1998), where a decomposition in base functions \(B(x, y)\) for the kernel was proposed:

\[github.com/toros-astro\]
\[ Ker = \sum k_i B_i(x, y), \text{ where } k_i \text{ are the coefficients. The image subtraction is carried by minimizing a cost function } Q: \]

\[ Q = \int_{(x,y)} [N(x,y) - (Ref(x,y) \otimes Ker(u,v))]^2 \]  

(3)

The first proposed kernel was a sum of Gaussian functions modulated by low order polynomials \((p_x(x) \text{ and } p_y(y))\), as the Eq. [4] where \(N\) denotes a normal distribution function.

\[ Ker(x,y) = \sum a_i N(\mu = 0, \sigma_u, \sigma_v) p_x(x) p_y(y) \]  

(4)

This model is not versatile enough for complex-structured PSFs, and the image subtractions performed with this methodology may present artifacts. Bramich [2008] proposed a more flexible model modification, that treats each pixel from the PSF as an independent value. This is equivalent to use a set of delta type basis functions (Eq. [5]), where each one is modulated by a coefficient which represents the pixel value located in position \((u_i, v_i)\).

\[ Ker(u,v) = \sum k_i \delta(u - u_i, v - v_i) \]  

(5)

The determination of \(k_i\) coefficients is performed during the minimization of the function \(Q\). These techniques have been applied before on various astronomical analyses such as variable star search, and exoplanet search, for example in Hoffmann et al. [2012]; Arellano Ferro et al. [2013]; Oelkers et al. [2013, 2015]; Lee et al. [2015]; Arellano Ferro et al. [2017]; Oelkers and Stassun [2018] just to name a few.

2.1.2. Zackay formal image treatments

The proposed image model by Zackay et al. [2016] is derived using statistical theory, as they choose to represent the pixels of the images as distributions, and attempt to perform hypothesis testing on their values.

The model for the reference image \(R\) is

\[ R(x,y) = F_R T \otimes P_R + \epsilon_R \]  

(6)

with \(T\) being the true image, i.e. taken with a perfect infinite telescope, and no atmosphere influence; \(F_R \in IR\) is the transparency, which incorporates the atmosphere and instrumental absorption, and would play the role of a flux zero point; the \(P_R\) is the PSF, which should be normalized to have unit sum; and \(\epsilon_R\) is the background noise with variance \(\text{Var}(\epsilon_R) = \sigma_{\epsilon_R}^2\), assumed to be normally distributed.

This model is suitable for statistical hypothesis testing of the existence of a new source, since it allows the definition of simple null and alternative hypotheses for the new image:

\[ H_0 : N = F_N \otimes P_N + \epsilon_N \]  

(7)

\[ H(q, \alpha) : N = F_N(T + \alpha \delta_q) \otimes P_N + \epsilon_N. \]  

(8)

The lack of evidence for the null hypothesis \(H_0\) favors the existence of a point source at position \(q\) with flux \(\alpha\) in the new image, which is affected by PSF \(F_N\), transparency \(F_N\) and noise \(\epsilon_N\). According to the Neyman-Pearson lemma (Neyman and Pearson [1933a]), the most powerful (following the definition given in Neyman and Pearson [1933b]) statistic is the likelihood ratio test:

\[ L(x,q) = \frac{P(N,R|H_0)}{P(N,R|H_1(x,q))} \]  

(9)

which can be calculated without prior information on \(T\). It can be proven that maximizing equation [9] is the same as maximizing the statistic \(S\) of equation [10]

\[ S := \frac{\log(L(x,q))}{\alpha} \]  

(10)

and after intermediate calculations available in the appendix A of Zackay and Ofek [2017b] and Zackay and Ofek [2017a] it is possible to prove that \(S\) is the cross match convolution of the real difference image \(D\) and its corresponding PSF \(P_D\), of equation [13]

\[ \tilde{S} = F_D \hat{D} \hat{P}_D \]  

(12)

By algebraic manipulation, \(D\) and \(P_D\) can be expressed as

\[ \hat{D} = \frac{F_R \hat{P}_R \hat{N} - F_R \hat{P}_N \hat{R}}{\sqrt{\sigma_{\epsilon_R}^2 F_R^2 \hat{P}_R^2 \hat{N}^2 + \sigma_{\epsilon_R}^2 F_R^2 \hat{P}_R^2 \hat{N}^2}} \]  

(13)

\[ \hat{P}_D = \frac{F_R \hat{P}_N \hat{R}}{F_D \sqrt{\sigma_{\epsilon_R}^2 F_R^2 \hat{P}_R^2 \hat{N}^2 + \sigma_{\epsilon_R}^2 F_R^2 \hat{P}_R^2 \hat{N}^2}} \]  

(14)

with the flux based zero point relative to the difference

\[ F_D = \frac{F_R F_N \hat{R}^2}{\sqrt{\sigma_{\epsilon_R}^2 F_R^2 + \sigma_{\epsilon_R}^2 F_R^2}} \]  

(15)

For source detection the authors claim that the best option is to determine the locations where \(S\) peaks above \(5\sigma\) (the robust \(\sigma\)). This is equivalent to a \(p\)-value test cut. We implemented this as a separate technique, as it is the source detection method originally proposed in the already cited work. Among other features, this \(S\) image statistic has correlated background noise, and thus is not suitable for the extraction of every astronomical information. To recover this Zackay et al. [2016] derives the formulation of an \(S_{corr}\) image in Eq. 98 from Appendix C, which is not affected by the noise level in the vicinity of bright sources and other additional noise components.

2.2. Elements of Machine Learning

DIA techniques provide the means to detect transient and variable sources on images. However, as several defects arise
and are detected as bogus, it is necessary to classify them in order to identify the real sources. Machine Learning (ML) takes advantage of the power of massive amounts of data to generate suitable models to assess the bogus real classification problem. In the classification of transient objects there are several implementations by big collaborations whose capability of collecting data got to overwhelm their human classification capacity. This situation forced them to innovate and apply several machine learning techniques to data selection, in order to make manageable their volumes of raw data, and focusing their attention on the most promising candidates. For a summary of the methodologies implemented in recent years see, e.g., Bloom et al. (2012).

2.2.1. Machine Learning Algorithms

Machine learning (ML) algorithms rely on the use of data to generalize the relations between intervening variables in order to make predictions. There are several classes of algorithms that belong to this area. They differ on how they generalize the examples, and more specifically, on the way they represent the data as models. A relevant review of learning with algorithms can be found in Domingos (2012), where a general discussion on ML related topics is carried out. The pertinent jargon adopted in this work can be summarized as:

- objective class: (also target class) the output variables that we want to predict. In binary classification problems they are commonly referred as “positive” and “negative” classes.
- instance: a data example. Can be a training (“labeled”) instance, or a new observation.
- features: the predictors, in short the measurable and/or computable quantities that represent each instance.
- score: a value used to quantify the performance of an algorithm to retrieving the objective class given a training-test dataset.
- confusion matrix: a double entry table where it is possible to visualize the results in terms of instances correctly and incorrectly classified. This table originates the score metric values used in class prediction problems.

For more details or references see, e.g. Mitchell (1997); Hastie et al. (2001). Training datasets are critical for supervised ML algorithms, which learn model representations designed to infer the objective class according to the describing features. A confidence score is used to select the best model representation. In this context, the training data must be labeled, using any previous classification available, such as human training. This validation process is objective at the time of understanding the results, and the meta-parameters of the model already trained provide second-order information on the data itself. In this work we employ supervised classification algorithms from the standard library Scikit-Learn (Pedregosa et al., 2011) —used version is 0.20.1—, which is one of the most popular libraries for ML, written in the convenient programming language Python.

Figure 1: Diagram exemplifying the combined DIA and ML analysis process developed in this work. Reference (R) and new (N) simulated images are processed using DIA by applying Alard and Lupton (H), Zackay et al. (Z), Bramich (B) methods, and $S_{corr}$ (S) image. We show for comparison, an example of a real object, detected from an injected transient source (TS) and an example of a bogus object arising from an artifact (Ar) in the image difference resulting of the application of the several DIA techniques used. In the bottom of this figure, we show three bars, separated into several blocks, each one representing different types of data. The first bar represents the results of source detection after the DIA algorithms have performed the image subtraction. The first two blocks of this bar represent the injected sources, which are divided into Missed and TS detected, and the left block is the resulting set of artifacts Ar. The ML algorithms attempt to learn the classification of TS and Ar subsets into Real and Bogus for each DIA technique. The results of ML are displayed in the second bar, where the sum of the blocks TS and Ar are split into Real and Bogus blocks of data. The combination of the DIA+ML results is represented in the last horizontal bar, showing the final quantities of False Negatives (FN), True Positives (TP), False Positives (FP) and the True Negatives (TN). As a final figure of merit we calculate the F1 for each combination of DIA+ML algorithms.

For more details or references see, e.g. Mitchell (1997); Hastie et al. (2001). Training datasets are critical for supervised ML algorithms, which learn model representations designed to infer the objective class according to the describing features. A confidence score is used to select the best model representation. In this context, the training data must be labeled, using any previous classification available, such as human training. This validation process is objective at the time of understanding the results, and the meta-parameters of the model already trained provide second-order information on the data itself. In this work we employ supervised classification algorithms from the standard library Scikit-Learn (Pedregosa et al., 2011) —used version is 0.20.1—, which is one of the most popular libraries for ML, written in the convenient programming language Python.
In what follows we summarize the ML algorithms used in this work.

- **K-Nearest Neighbors** ([Hastie et al., 2001]): a really simple algorithm, basically classifies an instance given the most popular kind in its vicinity on feature space, being K the number of close instances to consider for prediction a model parameter. Although this method has the advantage of fast training, it has several drawbacks. First of all, euclidean distance can be computationally quite expensive for high-dimensional spaces, and misleading compared to the real metric in feature space. Moreover, KNN requires keeping the training data stored to classify new instances.

- **Support Vector Machines**: a quite sophisticated algorithm. In a few words, this technique tries to characterize an hyperplane in the feature space that separates different classes. A major quality of this algorithm is that the procedure to find this hyperplane only depends on inner products of feature vectors (in the linear algebra sense), and so, non-linear transformation kernels can be used to make this classifier able to work in a wider range of problems.

- **Random Forest (RF)** from ([Breiman, 2001]): this is a so called meta algorithm since it is basically a combination of simpler ML methods. In this case RF is a collection of Decision Trees that uses a randomly chosen subset of features to train.

  A decision tree is a rather simple concept, basically is a chain of if--else that separates the data taking into account only one feature at the time. There exists several variations of this algorithm depending on the statistic that is used to separate the data, such as information gain, entropy maximization, etc. The Random Forest constructs an ensemble of trees, each of which casts a vote, and the majority decision is taken.

2.2.2. Feature selection process

Each feature provides different amounts of information about the objective class, with more informative features being the most important ones. To estimate this relative importance we can use a wide range of techniques that range from data visualization, to Principal Component Analysis ([PCA, Pearson (1901)])

Since some features might provide redundant information, it is convenient to obtain the maximum amount of information with the minimum set of features since it reduces dimensionality and pruning this non-informative variables can lead to an additional computational speed up. In some cases this can be achieved by a transformation of the feature space, by creating new computable features that reduce the dimensionality of the problem (e.g. PCA). Sometimes this is not necessary since it is possible to discard redundant features and just use the ones that have better performance.

In order to maximize the performance of the used ML algorithms we introduce convenient feature selection strategies for each case. The first strategy is to analyze importance for each feature individually, this is called univariate analysis. The simplest technique is just to filter features with low variance, since a constant quantity would hold no information regarding the target class. This general approach was applied to every tested algorithm. Another univariate technique is to calculate the mutual information between each feature and the target class ([Cover and Thomas, 2012]). Selecting features which maximize this value would in principle select the features with higher predictive capability on the target class. The mutual information technique was used for the KNN algorithm only.

For the decision tree family of algorithms we introduce the Random Forest derived feature importance calculation ([Strobl et al., 2007, 2008]). The analysis we used consists of training the model using every feature and in the testing stage carrying a random permutation of the values of each feature, erasing any correlation with the target class. The decrease in performance of the trained algorithm would quantify the importance of the permuted feature, without need of re-training the model. In order to determine the significance of this decrease we include in the training and testing set a control feature with random values, and compare the importances in relation to it. Any feature with an importance less than the random feature would be discarded. This technique is biased in the case of correlated variables, but this could be avoided by pruning them before performing the selection.

Lastly, in the case of SVM we employed a methodology known as Recursive Feature Elimination ([Guyon et al., 2002]), which works by using an external weighting algorithm, which is evaluated for random subsets of features, and recursively pruning those with low weight. The external weighting algorithm can be any linear model capable of delivering a coefficient for each feature, which makes this technique suitable for linear Support Vector Machines.

2.3. Evaluation of DIA+ML algorithms performance

The focus of this work is centered at the task of recovering transient sources from telescope images by combining the Difference Image Analysis and the Machine Learning methods in order to maximize recovery completeness and minimize its contamination. To test the training stage of a ML algorithm, a labeled testing dataset is used to generate predictions, and the performance can be quantified identically to a hypothesis test, by constructing the confusion matrix, as follows:

- **True positives (TP)** are the injected transient sources correctly detected and classified as real instances.
- **False positives (FP)** are the artifacts in the image differences and the misclassified instances of bogus objects.
- **True Negatives (TN)** are the correctly classified bogus objects.
- **False Negatives (FN)** are the positive instances lost due to misclassification or missed by the DIA.

Notice that the components of the confusion matrix can be computed for any detection and classification problem, in particular, either for the results of the DIA or of the DIA+ML, changing the previous definitions accordingly.
Using the values of the confusion matrix we can compute more sophisticated scores, useful to quantify performance metrics for different optimization strategies.

- **Precision** measures how many of the classified as positive instances were actually positive. It can be calculated like this: $TP/(TP + FP)$.

- **Recall (R)** or True Positive Rate (TPR) characterizes how many of the positive examples were actually retrieved. This is $TP/(TP + FN)$.

- **False Positive Rate (FPR)** is the probability of a false detection, this is $FP/(FP + TN)$.

- **False Negative Rate (FNR)** is defined as $1 – R$, and is the rate of lost positive examples.

Precision and recall are useful to check the algorithm’s performance in this unbalanced class context, where the bogus or artifact objects rate depends on the difference imaging method applied. A more informative value is the $F1$ score, derived from the precision and recall metrics, which correctly weights the cost of the errors of losing transients as well as detecting artifacts. $F1$-measure is the harmonic mean of the $P$ and $R$ metrics, both equally weighted, and can be used as a final figure of merit. It is computed using $2PR/(P + R)$, which is the same as $2TP/(2TP + FN + FP)$ and is also a number from 0 to 1. This metric is less sensitive to unbalanced classification scenarios, because it takes an intermediate value between $P$ and $R$, but staying closer to the lower value, penalizing the discrepancy of Precision and Recall. At the same time is a metric which does not require the value of the $TN$ amount, a quantity we cannot derive from any technique, due to the nature of the problem.

To assess the performance of a trained algorithm, usually new data is used. After the training stage, it is possible to detect cases of over or under training, by using labeled examples which were not processed yet. For this a standard technique named cross-validation is preferred, in which the same training set is divided into training and testing subsets. This allows us to record the misclassifications and build a confusion matrix. Therefore, we used **stratified** $k$-fold cross validation [Witten et al., 2016], which splits the training set in $k$ subsets: $k-1$ pieces serving for training and the remaining just for validation purpose. The results of the validation of this $k$ classification algorithms trained are the $k$ confusion matrices, one for each fold of test data. The several metrics explained above are then calculated, yielding a confident performance evaluation of the algorithm.

### 3. Simulated and real datasets

In order to test and compare the different combinations of DIA and ML techniques for transient detection, we explored a range of different observing conditions using a purpose-made dataset consisting of generated simulated images with injected transients. However these simulations are not completely realistic, so that we also tested the combined techniques using observations triggered by GW ALIGO alerts. In both cases the injection of transients allows us to assess suitable rate estimates to test the performance of the detection methods.

#### 3.1. Generating a simulated image dataset for ML training

We simulated images using [ASTROMATIC Software][1], particularly STUFF and SkyMAKER [Bertin, 2009], which together can produce realistic images of stars and galaxies, for any given telescope hardware configuration, and including image artifacts such as saturation, spikes, secondary mirror spider shadows, etc. Since it is open source software, it is possible to reproduce the results of the image simulation by introducing the same configurations. We simulated the data in several steps:

1. First we used STUFF to produce a catalog of real objects in the field, including galaxies and stars, containing their positions and real photometric properties, as well as shape parameters.
2. Then this catalog was used to make a fits image using SkyMAKER. This is taken as the "reference image" ($R$).
3. Next some stellar sources (transient) are added to the catalog previously created, at random positions, and with random magnitudes drawn from a fixed Luminosity Function (LF) distribution.
4. The final outcome is a "new image" ($N$) with the transients sources included.
5. The last stage of the simulation is to perform the DIA subtraction between $N$ and $R$, and perform the source detection on the resulting difference image.

These steps are repeated once for each point of the explored parameter space, having then, one $R$ and one $N$ image for each of them. The simulation parameter values cover a relevant range of possible observational configurations, taking into account three aspects, namely, the telescope & site characteristics, the sky stellar background, and the relative location and brightness of the transient with respect to their host galaxy. The simulations expand eight parameters, and each one has associated two images, one corresponds to the reference image and the other is the new image. The values of the parameters used in the images simulations are described in Table 1. Regarding the observational configurations, we considered five parameters, namely, the diameters of the telescope primary and secondary mirrors, the seeing FWHM for $R$ and also $N$, the plate scale, and the exposure time. The values of telescope apertures are selected so that they represent the available instruments by our collaboration. The seeing of the $R$ images took values of 0.8, 1, and 1.3 arc-seconds, following empirical determination of TOROS future site characteristic values (Fig. 3 in Renzi et al. (2009)). The seeing of the $N$ images took values of 1, 1.9, 2.5, motivated by typical and bad observing conditions. The plate scale and exposure times values are chosen according to the available CCD cameras. Regarding the telescope and site characteristics, we include as particular cases, the Estación

[1]: www.astromatic.net
Astrofísica de Bosque Alegre (EABA), the TOROS pilot instrument (TORITOS) and the projected 0.6-m telescope for the TOROS site.

The number and contrast of the stellar sources are described by the stellar density parameter. The range of stellar densities (given by the STARCOUNT\_ZP parameter) represents fields of different environments going from typical densities of an mid galactic latitude, and up to densities of less than 5 degrees from the MW disk center at $l \sim 60$ degree of longitude, with a limiting magnitude of $i \sim 19$. The luminosity distribution of these sources are governed by a power law, with an exponent which took the values of 0.1, 0.5, 0.9 dex per mag (STARCOUNT\_SLOPE parameter in the SKYMAKER software).

Also, we allow the variation of the background surface brightness, using values of 20 and 21 for reference images, typically taken mostly on dark nights, and 20, 19, and even 18 for new images, taken in different conditions. Regarding the host galaxies of the injected transients, we sampled the relative brightness and the angular distance from the host center from uniform distribution in the ranges [-4, 1] magnitudes and [0.5] half light radius, respectively. This allows us to explore different transient/host relative configurations, including low relative luminosities and position ranging from the center up to the outer stellar halo.

For a given observational configuration (or set of parameters), we define the magnitude range where reliable photometry can be obtained, based on the photometric calibration obtained applying RANSAC (Fischler and Bolles, 1981) robust linear regression on standard sources. The RANSAC method prunes spurious sources, and obtains an estimation of slope and zero point values, not sensitive to outliers, and at the same time also provides a filter mask identifying this outliers.

The explored parameter space may include configurations which are not probable. For instance the combination of a 1.54 mirror, with 300 seconds exposure time, in a night with a bright background light (i.e. moon light), a large plate scale and a broad seeing. A number of this corner cases are present in the explored configuration space, and in some of this cases simulated images appear completely saturated, and in others the photometric quality is extremely low. This corner cases have been discarded, leaving a total of 26205 groups of $N$, $R$ and DIA differentiated images. The results shown in the next sections include all sensible points in the explored parameter space, which comprises an heterogeneous combination of image qualities. Nevertheless, the independent photometric calibration of each image informs us the range of validity of flux determination on each configuration making us able to fairly compare results among the whole simulated dataset.

In section 3.2 we discuss the general trends that result from our analysis. Although our parameter space do not cover all possible configurations for telescope optics and site, we provide the codes that allow to simulate any other observational configuration.

A total of 3272784 transients were injected, placed on top of an extended object (as expected in the case of Kilonovae) with a random angular position and distance relative to the host galaxies less than 5 half light radii. The simulated galaxies have different morphological types, and also have different redshift values, random orientation and ellipticity and their luminosities are chosen according to a Schechter luminosity function. The $R$ magnitudes of the transient objects are calculated with a random offset from the host galaxy, drawn from a uniform distribution, between values -4 and +1 magnitudes.

In the Fig. 1 we present a scheme of the results of the difference image subtraction and the posterior ML classification of real and bogus transient sources. Besides the reference and new simulated images we show the result of the subtractions performed for this pair of images.

The subtractions were carried out by three different implementations of the techniques introduced above, namely: Zackay, Alard & Lupton, and Bramich algorithms, and we show stamps of detected transient sources and artifact objects for visual comparison. We also present the resulting $S_{core}$ image computed as in Eq. 98 from Appendix C of Zackay et al. (2016).

For the implementation of the Zackay and Alard & Lupton method we adopted the publicly available HOTPANTS software by Becker (2015) (version 5.1.11). The Bramich implementation is a Python code by the authors, available at github.com/toros-astro/cis (version 0.1.14), as well as Zackay et al. implementation, which is also a Python code by the authors available at github.com/toros-astro/ProperImage. Both implementations are built upon standard scientific libraries such as NumPy (version used here is 1.15.14), SciPy Jones et al. (2001–) (version 1.1.0) and Astropy (Astropy Collaboration et al. 2013) Price-Whelan et al. 2018 –version 3.0.4– and run on versions of Python 2.7 as well as 3.6. Also cis and Properimage are fully documented and tested, and they are Open Source, free to the community to use. Many examples and details of the implementation can be found in the documentation at optimal-image-subtraction.readthedocs.io and properimage.readthedocs.io. It is worth noticing that the Zackay implementation works faster when the exposure times of the reference and new images are equal. In this case there is almost no need for a zero point calibration, which saves computational time. The Alard-Lupton implementation is written in C programming language, also it employs a simpler Gaussian PSF assumption, making this method faster than the others by a factor of almost 4X.

3.2. Injection of transient objects in observed images

The simulated images previously used are practical for developing the DIA techniques and generating a training dataset for the bogus/real classification problem. Also this approach allows us to explore the dependence of the algorithm performance on different observing settings and transient properties. Nevertheless, this approach is limited by the simplifying hypothesis. In order to take into account the flaws that arise in the observing process and subsequent analysis, we present in this subsection the process of injecting transients sources into real observational images.

We used images obtained by the TOROS collaboration as part of the follow up of the triggers during Advanced LIGO.
science run O2. The images were obtained for the gravitational wave event GW170104, using the Estación Astrofísica Bosque Alegre (EABA) 1.54-m Newtonian telescope. The instrument was set to white light image acquisition, and the CCD used was an Apogee Alta U16. Since the observations were performed hours after the gravitational wave event trigger was received, we had no previous references of the selected targets. The “reference a posteriori” methodology was implemented, using images taken over the following months. The objects were selected by cross-correlating and filtering the skymap provided by LSC GCN:20364 and the galaxy catalog from White et al. (2011), according to the methodology described in the first TOROS follow up paper (Díaz and Collaboration, 2016). The following galaxies were observed at two epochs separated by eleven months: NGC1341, NGC1567, NGC1808, ESO0555-022, ESO3564-014, PGC073926, PGC147285. The images were reduced using a standard image processing, and then co-added for each epoch using the SWarp (Bertin, 2010) public software.

The procedure for the subtraction analysis is consistent with the one performed on the simulated dataset. We perform the subtractions using each image twice, once as a reference, and once as a new image. Given the BH-BH merger nature of the GW event (Abbott et al., 2017b) no EM emission was expected from this source, and in fact no kilonovae were detected previously introduced. These metrics, among others, were used to rank the transient detection techniques, thus allowing us to chose the most suitable approach for applications on TOROS images.

| PARAMETER                          | UNITS     | VALUES                  | TOROS instruments |
|-----------------------------------|-----------|-------------------------|-------------------|
| aperture of the telescope         | [m]       | [0.4, 0.6, 1.54]        | 1.54 0.6 0.4      |
| reference seeing FWHM             | [arcsec]  | [0.8, 1, 1.3]          | 1.3 0.8 0.8       |
| new image seeing FWHM             | [arcsec]  | [1.3, 1.9, 2.5]        | 2.5 1.0 1.0       |
| plate scale                        | [arcsec/pix] | [0.3, 0.7, 1.4]   | 60, 120, 300      |
| exposure times                     | [sec]     |                         |                   |
| stellar density                    | [stars per sq deg] | [4e3, 8e3, 32e3, 64e3, 128e3, 256e3] |                   |
| stellar luminosity distribution exponent | [dexp per mag] | [0.1, 0.5, 0.9] |                   |
| background brightness (R)          | [mag per arcsec ] | [20, 21, 22]        |                   |
| background brightness (N)          | [mag per arcsec ] | [18, 19, 20]        |                   |
| relative brightness from host      | r-band magnitudes | sampled from Unif(-4,1) |                   |
| angular distance from host         | half light radius | sampled from Unif(0, 5) |                   |

Table 1: Parameter space for simulated images to be explored for transient detection.

4. Comparing transient detection agents

The DIA methods can deliver either direct subtractions (those which return an astronomical image without any convolution process), or convolved subtractions, as the $S$ statistic proposed by Zackay et al.. For the direct difference image methods, the transient source candidate detections were performed using SExtractor (version 2.19.5), with the same configuration parameters.

The $S$ image, in turn, is a cross-correlation of the difference image with its own PSF, and so the detection of transient candidates is different from the other DIA methodologies. By calculating the $S_{corr}$ statistic from the $S$ image and performing a robust determination of its mean and standard deviation ($\mu$, $\sigma$), we find the significant sources by thresholding the pixels:

$$S_{corr} = (S_{corr} - \mu)/\sigma > \alpha (= 3.5).$$

In Fig it can be seen the distribution of significances for the artifacts and transient sources as well as both cuts of 3.5 and 5 in vertical lines, for both datasets. The chosen threshold of 3.5
is relaxed with respect to the value of 5 originally proposed by Zackay et al. in an attempt to detect dim transient sources. Although this increases the number of artifacts, the posterior ML analysis is expected to label them as bogus. Since SExtractor performs a pre-convolution on every image it scans, we cannot use it on $S_{corr}$ images. Instead we obtained candidates on the $S_{corr}$ images using sep, an open source package capable of running a source extraction without kernel pre-convolution. This software provides photometric measurements similar to those delivered by SExtractor. The threshold used with this source detection technique is again a limit on 3.5 over the background.

Therefore we were able to analyze the candidate sources, their photometric properties and parameters measured by SExtractor and sep for every DIA technique, and use this results as features for the ML stage. We end having five DIA transient detection results: detections over Zackay D images (Z), over Bramich (B) and over Alard-Lupton differences (A), the three of them provided by SExtractor; detections over $S_{corr}$ using sep ($S_{sep}$), and detections using a simple pixel treshold ($S_{3.5\sigma}$). In what follows (Subsec. 4.1), we explore the main differences between samples of artifacts, real transient sources and missed sources for each DIA method. Since the comparison is based on the very same images, we can directly relate the occurrence of both real transient sources and missed objects across methods, although this is not possible in the case of artifacts, since those can be random subtraction errors misidentified as transient candidates. The labeled candidates for Z, B, A and $S_{sep}$ DIA results, and their photometric quantities measured are the inputs of the feature selection process, for training and testing of the ML models aimed at doing the bogus–real classification (Subsec. 4.3).

### 4.1. Performance of the DIA methods

The fraction of occurrences of the different classes of objects is our first piece of information regarding the subtraction methods performance, prior to any Machine Learning technique application. In Table 2 we show the number of transient sources (TS), missed detections (missed) and artifact sources (Ar), along with the corresponding fractions, TPR, FNR and FPR, respectively. We also report the F1 statistic after the DIA implementation that will be later compared to the same statistic after the ML application.

By definition the values of FNR and TNR always add up to one, and FPR can be any number, since the normalization is over the total number of injected sources. Regarding the rates of recovered transient sources (TPR), we read that there is variability on the results for different techniques. There is a baseline of 50% for every technique, and a top value of 93%, finding intermediate values in the simulated as well in the real dataset. The number of missed objects is larger in the case of the simulation, since we injected transients in a larger number of simulated images, covering a wide range of experimental configurations (as indicated in Table 1).

For the simulated dataset we can read that Bramich finds less transients, and at the same time produces less artifacts. For the real dataset it finds more transients than any other technique, and produces a relatively low amount of artifacts. The technique which finds more transient sources is $S_{sep}$, with a relatively low number of artifacts for the simulated dataset. In the case of the real dataset the scenario is the opposite. The method which generates more artifacts in the simulated case is Alard-Lupton, yielding more than twice the amount of false detections than Zackay technique. In case of the real dataset this happens for the $S_{3.5\sigma}$ which has a FPR of 200, and Zackay generates less artifacts than any other DIA method. This behavior can be explained with the tendency of $S_{3.5\sigma}$ of finding local maxima in the edges of images, or near bright sources, an issue we would like to address in the future.

In the simulated dataset we find that the $F1$ statistic is systematically higher than the real dataset results. This is mostly due to extremely high FPR we measure in the latter. It is clear that we have more sources of confusion in the real images, this could explained by the presence of instrumental defects on the CCD camera used, poor flat field calibration, and correlated noise coming from the stacking procedure. These effects are not straightforward to include in the simulations therefore we can think of them as a representation of an optimistic case sce-
nario, in comparison to the observations.

### 4.2. Analyses of the DIA results

In order to compare the photometric properties of the transient candidates for every DIA method, we performed a photometric calibration with the flux of the simulated astronomical sources, by using a robust linear regression as already detailed in Sec. 3. The measured magnitudes of the transients recovered in the simulation generated images show agreement among the different DIA methods. We show in the Fig. 3 the mean and the standard error of the difference between the injected and measured magnitudes of the transients recovered in the simulated images, as a function of the transient r-magnitude. We also show in this Figure the results for transients injected on EABA observations, and the subset of simulated images that are closest to the observing configuration of the EABA telescope and site. The difference between the EABA observations and the corresponding simulation arises because of several factors. Most importantly is that the real dataset images are stacks of images, and this largely enhances its dynamic range, allowing to observe bright sources without saturation, and at the same time, sources at magnitudes fainter than 17 with an good signal to noise ratio. In the simulations, the aperture photometry might not be able to ideally capture the true flux value of the bright saturated or enlarged sources. Prior to reaching the saturation limit, the linearity of the CCD response is lost, progressively deviating flux measurements from true values. The limiting magnitude difference between simulations and real data are due to the actual performance of the EABA instrument, and the atmospheric conditions during the nights of O2 follow up observations. Taking this into account, it is worth noticing that the simulated and observed images are consistent within a range of approximately 5 magnitudes.

We also studied several statistics for each class of object, trying to gain insight into their properties. The Cumulative Luminosity Function for each class in the simulated dataset is displayed on the top row of the Fig. 4, where it can be seen that for the transient sources the methods are roughly equivalent. We find that for the magnitude range $r \geq 15$ the methods behave similarly, but Zackay and S-Corr have more transient sources detected than the rest of the techniques. In this faint end of the luminosity function the main reason for losing objects is the detection limits. In the bright end $r \leq 15$ however, there is not a clear pattern, and besides the true missed objects, the already discussed errors in the magnitude measurements for bright objects due to saturation could increase the discrepancy with the simulated magnitude values.

In the case of the artifacts, both in the simulated and observed dataset, we find similar behavior in the DIA techniques, although it is worth to notice that the accumulated number of objects is in disagree with the reported figures in Tab. 2, this is due to the fact that many artifact sources present flux measurements without astrophysical meaning. The values of magnitudes calculated for a portion of the artifacts fall outside the range presented in the Fig. 4.

The right panel of Fig. 4 top row, corresponding to the missed objects in the simulations, shows that all methods fail to recover an important number of sources fainter than 21 magnitudes, and also objects brighter than 12 magnitudes. Since this missed, and transient sources sets of objects are the complement of each other we find similar explanations for the bimodality of lost sources and the deficits of injected recovered objects. Particularly S-Corr is the method which lose less injected sources, followed by Alard-Lupton, although the latter losses objects in the whole range presented, in contrast with the other methods. As a comparison with the real dataset, we find that there is not such gap in between the faint and bright end of the missed objects luminosity function, but instead a smooth ever increasing distribution is observed. In the real dataset, we have a different picture, with every method losing objects, almost equally, except for the Bramich method, whose distribution seems to be fainter, and shows accumulated numbers below every other technique. We also observe a slight increase in the number of objects lost for the S-Corr compared to the rest of the techniques, although is should be recalled that there is a big gap between the sample sizes of simulated and real datasets.

| Table 2: Number of occurrences of transient sources (TS), missed detections (missed) and artifact sources (Ar), and the corresponding rates (TPR, FNR, FPR, F1). |
|----------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|
|      | TS      | Missed | Ar      | TPR    | FNR    | FPR    | F1     |
| Z    | 1933065 | 1339719| 3170089 | 0.59   | 0.41   | 0.97   | 0.46   |
| B    | 1596713 | 1676071| 1979948 | 0.49   | 0.51   | 0.60   | 0.47   |
| A    | 1971291 | 1301493| 5537472 | 0.60   | 0.40   | 1.70   | 0.37   |
| S_{\text{sep}} | 2180390 | 1092394| 2456876 | 0.67   | 0.33   | 0.75   | 0.55   |
| S_{3\text{Sr}}  | 2092625 | 1180159| 2700107 | 0.64   | 0.36   | 0.83   | 0.52   |

Figure 3: Magnitude difference between injected and recovered sources as a function of the magnitude of the injected transients, for the full averaged simulation (thick light gray), the stacked EABA images (circles) and a simulation with an equivalent mirror size and exposure time (solid black). Error bars are $1 \times \sigma$ wide.
Figure 4: Cumulative Luminosity functions for Transient Sources (TS) (left column), Artifacts (Ar) (center column) and Misses (right column) object classes. In the first row we have the simulated data and in the second row the real dataset. The cumulative luminosity function of the injected sources is displayed in every panel as the shaded area. The line code for DIA techniques is: Zackay’s is always in lines with white dots, where Bramich’s is in lines with filled dots, Alard-Lupton is in dotted lines, as well as $S_{core}$ statistics is in black solid line.
In some cases a higher contamination of bogus may be acceptable since Machine Learning algorithms could separate them from the true interesting candidates, while on the other hand, a high FNR is quite undesirable since those candidates are “lost forever”. This compromise can be optimized by using supplementary techniques, for instance by requiring that any transient candidate to be persistent during at least two consecutive images during the observing night, this would have impact by creating a more favorable ratios scenario.

4.3. Machine Learning Results

As explored above, we have photometric properties for the detected candidates in the several DIA images. On top of this we also have shape properties, as well as high order statistical moments on their light distributions provided by SExtractor -an in the case of $S_{corr, sep}$. This labeled dataset together with the mentioned features, constitutes the input instances for training and testing ML algorithms in the task of classifying the bogus and real objects.

In order to assess the different scenarios we simulated, our machine learning experiments were conducted in several steps:

- We grouped the dataset in terms of three simulation configurations values: the mirror diameter of the telescope, the exposure time, and the seeing of the new image. This gave us 27 subsets of data, where we conducted identical and independent experiments.

- Each experiment was carried out firstly by splitting the dataset into a training and final testing set, with a 20%-80% proportion respectively, due to the enormous amount of data available for training.

- The training subset is used to perform feature preprocess and selection, and to perform a k-fold cross validation performance measure for three ML algorithms: k-Nearest Neighbors, Random Forests, and linear Support Vector Machines.

- We calculated the confusion matrix for the ML classification (Bogus–Real), and combined it with the DIA performance metrics, in order to rank the DIA+ML methodologies using an overall figure of merit. This is done by deriving a confusion matrix from the injected sources, through the DIA (Missed, TS, Artifacts), to its final DIA+ML classification results (FN, TP, FP, and TN), such as illustrated in Fig. 8.

The used Machine Learning algorithms as previously stated were:

- k-Nearest Neighbors, using 7 neighbors, with uniform weights and euclidean distances (using scaled feature values).

- Random Forest, with 800 trees, with up to 7 features per tree, stopping the tree growing if less than 20 examples per leaf, using a Gini impurity criterion.

- Support Vector Machines, using L2 norm penalization, with a tolerance parameter of $10^{-5}$, solving the dual optimization problem, and weighting the classes if unbalanced for their frequency.

All the configurations for the ML algorithms correspond to Sci-Kit Learn version 0.20.1.

4.3.1. The feature selection process

We performed a preprocessing of the features, by scaling them to zero mean, and unit variance. Afterwards we applied univariated analysis by using a variance threshold cut of 0.1, pruning constant features. Following this simple treatment we used three different feature selection strategies, adapted to each ML algorithm tested.

To calculate the importances for kNN algorithm, we used the mutual information of the features and the target class, selecting the percentile 30 as a threshold cut.

For RandomForest we applied a feature selection process following Bloom et al. (2012), using a RandomForest training stage, and picking those features that were the most informative in the majority of the individual trees. To avoid bias in the selection we pruned the correlated features, and afterwards we followed the methodology described in associated Python package rfimpin a 10-fold cross validation experiment. To determine the unimportant features we added to the training dataset a uniformly distributed random variable. Using this procedure we can set the zero value of the scale as the importance of this Random feature. We calculated the mean and standard deviation of the 10 values obtained in the 10-fold experiment, tossing away those features consistent with the values obtained for the Random one.

For Support Vector Machines we applied a Recursive Feature Elimination on a 6-fold experiment, using an elimination step of 1, and choosing F1 as the scoring metric to maximize.

4.3.2. Evaluation of DIA+ML algorithms

In Fig. 5 we show a heatmap of values of F1 statistic (scaled by a factor of $10^4$) for the results of the 12 DIA+ML combinations (4 DIA techniques and 3 ML algorithms). The map is obtained by grouping several possible instrumental configurations, spanning the dimensions of the FWHM for the new image ($N_{FWHM}$), measured in arcseconds, the exposure time ($t_{exp}$) measured in seconds, and the diameter of the primary mirror ($D$) in cm, and performing the ML train–test experiments on each group. This covers a wide range of instruments, from small to medium size telescopes, and from good to poor observing conditions. There exists though, several possible configurations which are not covered by our analysis, designed mostly for TOROS collaboration available instruments. Nevertheless the analysis is valid for numerous transient search science collaborations with instruments falling within the range of our simulation parameters, such as (BackGEM [Ghosh and Nelemans 2015], ASASSN [Shappee et al. 2014], Catalina Sky Survey [Djorgovski et al. 2011], to name a few). It is also worth noticing that for each of the 27 instrumental configurations analyzed...
we are mixing the combination of values of the rest of the simulation parameters (listed in Tab. 1), and in consequence including several dissimilar transient detection scenarios into the same ML experiment. This result shows an expected dependence in the simulation parameters, clearly favoring longer exposure times and smaller seeing FWHM for the new image, there is also weaker but present dependence on mirror size. The strongest dependence of the $F_1$ value holds with the DIA technique applied. It is clear that independently of the ML used Alard & Lupton DIA method has better performance in terms of $F_1$ statistic in the simulations, followed by Zackay’s techniques (including $S_{corr}$). There is no major difference across ML algorithms for a given DIA technique, but a small advantage seems to be obtained when using RandomForest.

We also include the values of $F_1$ for the real dataset in the top of the map, which can be compared to the highlighted equivalent simulation. In the observed dataset we find that generally Bramich is better ranked, and the best DIA+ML technique is its combination with kNN. Notice that the overall $F_1$ for this DIA method is much better in this observed dataset than in the simulations. However the range of values for the DIA+ML combinations are comparable, despite the difference in the $F_1$ values of DIA only.

In order to better compare the performances reported in Fig. 5 we have marginalized over the groups of instrumental configurations, showing the quartiles of the distribution of $F_1$ values in the Fig. 6. In order to compare these values before and after ML classification we also include as horizontal lines the quartiles of the distribution of the $F_1$ values obtained in each group after the application of the DIA methods only. It is clear from this figure that Alard&Lupton technique is better ranked than any other DIA method, and at the same it experiences the largest boost in performance after the application of any ML algorithm.

As a global result we report that the best combination of DIA+ML for the simulated dataset is Alard & Lupton implementation Hotpants, and RandomForest. For the real dataset in turn we find that the best performance is with Bramich DIA combined with kNN machine learning algorithm. For these two cases we present the selected features and their normalized relative importances in Fig. 7. The included features in this figure sum up 90% of the total importance calculated for the whole set of selected features in each DIA+ML case. We can see that in the simulation the 5 most important fea-

| EABA: | 746 704 668 840 813 680 775 719 621 585 506 450 |
|-------|---------------------------|
| 400   | 629 627 629 597 593 634 636 638 731 742 738 |
| 600   | 637 643 643 599 605 606 648 660 660 734 743 737 |
| 1540  | 655 658 658 614 623 624 679 681 681 739 746 741 |
| 120   | 715 738 734 640 644 645 818 843 832 732 737 733 |
| 600   | 734 745 742 653 668 665 835 844 832 737 758 750 |
| 1540  | 738 756 752 667 683 683 817 848 838 742 754 748 |
| 120   | 574 796 791 683 705 703 760 805 791 774 787 782 |
| 1540  | 781 802 798 716 720 721 777 818 805 789 795 790 |
| 600   | 564 577 577 511 527 528 583 594 595 688 684 683 |
| 1540  | 589 594 595 534 557 558 616 626 627 699 704 701 |
| 120   | 663 695 691 592 605 604 814 839 827 856 702 697 |
| 1540  | 671 704 699 599 617 617 807 831 821 902 706 701 |
| 400   | 683 712 707 614 637 636 797 843 833 696 715 709 |
| 1540  | 713 745 738 622 632 636 764 809 799 720 737 733 |
| 300   | 706 740 735 629 655 657 771 812 801 728 736 733 |
| 1540  | 716 755 749 650 678 678 772 823 810 734 745 739 |
| 600   | 501 512 513 442 459 462 532 541 542 636 639 638 |
| 1540  | 523 534 533 463 472 476 566 576 576 654 663 660 |
| 400   | 619 648 644 540 561 562 793 820 808 556 664 660 |
| 1540  | 628 656 651 539 574 572 784 812 799 667 665 661 |
| 400   | 639 663 657 552 580 580 780 828 819 673 673 668 |
| 1540  | 649 684 678 555 600 603 756 804 797 680 696 693 |
| 300   | 650 683 679 574 616 615 743 802 781 688 690 687 |
| 600   | 652 690 686 580 611 613 742 793 784 691 694 691 |

Figure 5: A heatmap of $F_1 \times 10^3$ score performance values for each of the 27 instrumental configurations (as rows) where the DIA+ML algorithms have been applied (as columns). In the top row the scores for the observed dataset are included in the same grayscale, and a separated row is highlighted for comparison, corresponding to the simulations with values of $(N_{FWHM}, t_{exp} = 60s, D = 1.54)$. 

$13$
fures in the case of the best ranked DIA+ML, for the simulated dataset, that is A&Lupton, are in order ELONGATION, FLAGS, FWHM_IMAGE, MAG_AUTO and CLASS_STAR. In case of the real dataset, for Bramich+kNN the 5 most important features are in order B_IMAGE, CXX_IMAGE, CYY_IMAGE, SN, FLUXERR_ISO. The details on the calculation and meaning of each feature are included in Appendix A. The disagreement among the maps of importance might be explained considering the differences in the DIA+ML algorithm, and its feature selection procedure. Still, we can detect that the most important features represent as expected, shape and brightness parameters.

In order to gain some insights on the simulated dataset, we explored some of the dependences of the metrics of performance with the parameters of the injections. In the Fig. 8 we show the values of Recall R and F1 metric scores, as two dimensional maps, for two pairs of the simulation parameters, for the A&Lupton+RandomForest technique. The first one, in left panel, shows the dependence of $R$ with the values of relative position and brightness to the host galaxy. This maps clearly shows a strong dependence of the Recall of the final DIA+ML on the relative brightness, and at the same time shows no dependence on the distance to the host galaxy center. The second map, in right panel, shows the F1 values as a function of the parameters of the stellar luminosity function of the star field on the images: STARCOUNT_SLOPE and STARCOUNT_ZP. As explained in Sec. 3.1 the stellar luminosity function is a power law, and the exponent is the value of STARCOUNT_SLOPE, the STARCOUNT_ZP parameter is the total density of stars in the field (see Tab. 1). This map indicates that in a dense environment the F1 is lower, indicating the number of stars in the image as source of confusion for the DIA+ML technique. At the same time a lower slope, which traduces into in more bright stars at a fixed total density, pushes the scoring to lower values. It is clear then that dense stellar fields, like the ones in the galactic plane, are places where a decrease in the performance of the DIA+ML is expected. These class of studies are possible to evaluate by using a multi parameter simulation, like the one carried out.

5. Conclusions

We developed open source tools for image subtraction following the DIA techniques from Bramich and Zackay et al. We also made use of the HOTPANTS implementation of Aaldr-Lupton algorithm. The Python packages developed, together with HOTPANTS, were mounted on top of a CORRAL data processing pipeline, in order to systematically perform image subtractions with transient candidate injections. The images used for this were drawn from data obtained during an EM counterpart search, carried out by TOROS during O2 LIGO-Virgo-Scientific collaboration science observing run in 2017. These observations were performed during January and November, using the “reference a posteriori” methodology previously applied by the TOROS collaboration. Additionally we developed comprehensive simulations of images, exploring a multi dimensional parameter space of instrumental configuration as well as observing conditions, as detailed in Tab. 1 generating millions of transient injected on top of extended sources over several thousands of images. The nature of the injected transients does not include moving objects, or stellar variability, focusing the analysis on Kilonova/Supernova detection scenarios.

The mentioned pipelines measured the ratios of recovery of injected transients, as well the source contamination and loss for each DIA algorithm. We also compared their photometric results, including the $S_{corr}$ image photometry. In order to separate the true transient candidates from the spurious artifact detections we applied Machine Learning algorithms, trained with data generated by our pipeline. We carried out a feature selection stage, and completed a cross validation train-test experiment, in order to calculate score metrics such as Precision and Recall, useful to compare performances in an unbalanced class context, and ranked the methodologies by combining them in the F1 statistic.

The comparison of the simulated and real data showed that the scenario was of relevance in the performance of the different combination of methodologies, highlighting differences which can relate to the techniques as well as the nature of data used. Our results shows that Zackay’s image subtraction techniques, including $S_{corr}$, are more suitable for transient detection as a standalone technique. However it is clear that the Machine Learning algorithms are key to complete the task of selecting the relevant transient candidates, by setting apart the artifacts which contaminate and are uninteresting. After the application of these algorithms, and looking at the final performance metric F1 we conclude that, among the DIA+ML combinations tested, the better ranked technique were: in the real EABA images dataset Bramich+kNN, and in the simulated dataset A&Lupton+RandomForest. For these selected methods we also report the more important features, determined by Random Forest permutation importance in the case of the simulations, and by univariate analysis in the case of the real dataset.

Seizing the large and complex simulation generated, we analyzed the dependency of the Recall metric on the environment of the injected transients, in relation to the host galaxy, and the stellar field present in the simulated images. Concluding that
Figure 7: Radial plot of normalized feature importances. The left panel corresponds to the feature selection for the RandomForest algorithm in the simulated dataset. The right panel corresponds to the feature selection specific to the kNN algorithm as explained in Sec. 2.2.2, in the real dataset. The details on the calculation and meaning of each feature are included in Appendix A. The shaded areas correspond to the better ranked DIA+ML technique.

is more likely to generate artifacts and miss a fraction of transient objects in a dense and bright stellar field, and also to miss a greater fraction of detections if the contrast in brightness with the host galaxy is small, independently of its spatial relative location.

Software for DIA analysis pipeline, as well as other interesting astrophysical codes used in this research are available through the toros-astro github repository: [github.com/toros-astro](https://github.com/toros-astro). The image generation codes and ML experiments can be found at [github.com/BrunoSanchez/RBpipeline](https://github.com/BrunoSanchez/RBpipeline) and [github.com/BrunoSanchez/RBMLeaning](https://github.com/BrunoSanchez/RBMLeaning). Simulations results and data used in this work are available to the community upon request.

These results are very important for the development of the data processing pipeline of the TOROS collaboration which comprises different telescopes and instruments. A future extension of this work is to tackle the genuine transient time-series astrophysical classification for large amounts of data.

Acknowledgements

This work was partially supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina) and the Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba (SeCyT-UNC, Argentina). M.B and M.D. acknowledge NSF support through grant NSF-HRD 1242090. JLN is grateful for financial support received from the GRANT PROGRAMS FA9550-15-1-0167 and FA9550-18-1-0018 of the Southern Office of Aerospace Research and development (SOARD), a branch of the Air Force Office of the Scientific Research International Office of the United States (AFOSR/IO).

This research has made use of the `adsabs.harvard.edu` repository, the SIMBAD database, operated at CDS, Strasbourg, France. Also the Python programming language and the following scientific libraries: Numpy, Scipy, Astropy, scikit-Learn, and rfpimp.

References

Abbott, B.P., Abbott, R., Abbott, T.D., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., 2017a. Multi-messenger observations of a binary neutron star merger. ApJ 848, L12. URL: [http://stacks.iop.org/2041-8205/848/i=2/a=L12](http://stacks.iop.org/2041-8205/848/i=2/a=L12)

Abbott, B.P., Abbott, R., Abbott, T.D., Acernese, F., Ackley, K., Adams, C., Adams, T.e.a. (LIGO Scientific and Virgo Collaboration), 2017b. Gw170104: Observation of a 50-solar-mass binary black hole coalescence at redshift 0.2. Phys. Rev. Lett. 118, 221101. URL: [https://link.aps.org/doi/10.1103/PhysRevLett.118.221101](https://link.aps.org/doi/10.1103/PhysRevLett.118.221101)

Abbott, B.P, Collaboration, T.L.S (LIGO Scientific Collaboration and Virgo Collaboration), 2016. Observation of gravitational waves from a binary black hole merger. Phys. Rev. Lett. 116, 061102. URL: [http://link.aps.org/doi/10.1103/PhysRevLett.116.061102](http://link.aps.org/doi/10.1103/PhysRevLett.116.061102)

Alard, C., Lupton, R.H., 1998: A Method for Optimal Image Subtraction. ApJ 503, 325–331, doi:10.1086/305884, [arXiv:astro-ph/9712287](http://arxiv.org/abs/astro-ph/9712287)

Arellano Ferro, A., Bramich, D.M., Giridhar, S., 2017. CCD time-series photometry of variable stars in globular clusters and the metallicity dependence of the horizontal branch luminosity. Rev. Mexicana Astron. Astrofis. 53, 121–131, [arXiv:1701.02719](http://arxiv.org/abs/1701.02719)

Arellano Ferro, A., Bramich, D.M., Giridhar, S., Figuera Jaimes, R., Kains, N., Kuppuswamy, K., 2013. Variable Stars in the Globular Cluster NGC 288: [Fe/H] and Distance. Acta Astron. 63, 429–451, [arXiv:1401.5476](http://arxiv.org/abs/1401.5476)
Renzi, V. et al., 2009. Caracterización astronómica del sitio cordón macón en la provincia de salta. Boletín de la Asociación Argentina de Astronomía 52, 285–288.

Sanchez, B., Beroiz, M., Diaz, M., Macri, L., M., D., 2018. No bh em emission in prep.

Sedaghat, N., Mahabal, A., 2017. Effective Image Differencing for Real-time Transient Hunting. ArXiv e-prints [arXiv:1710.01422].

Shappee, B. J. et al., 2014. The Man behind the Curtain: X-Rays Drive the UV through NIR Variability in the 2013 Active Galactic Nucleus Outburst in NGC 2617. ApJ 788, 48. doi:10.1088/0004-637X/788/1/48

Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A., 2008. Conditional variable importance for random forests. BMC Bioinformatics 9, 307. URL: https://doi.org/10.1186/1471-2105-9-307

Strobl, C., Boulesteix, A.L., Zeileis, A., Hothorn, T., 2007. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics 8, 25. URL: https://doi.org/10.1186/1471-2105-8-25

White, D.J., Daw, E.J., Dhillon, V.S., 2011. A list of galaxies for gravitational wave searches. Classical and Quantum Gravity 28, 085016. URL: http://stacks.iop.org/0264-9381/28/i=8/a=085016

Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kauffman.

Zackay, B., Ofek, E.O., 2017a. How to COADD Images. I. Optimal Source Detection and Photometry of Point Sources Using Ensembles of Images. ApJ 836, 187. doi:10.3847/1538-4357/836/2/187 [arXiv:1512.06872]

Zackay, B., Ofek, E.O., 2017b. How to COADD Images. II. A Coaddition Image that is Optimal for Any Purpose in the Background-dominated Noise Limit. ApJ 836, 188. doi:10.3847/1538-4357/836/2/188 [arXiv:1512.06879]

Zackay, B., Ofek, E.O., Gal-Yam, A., 2016. Proper Image Subtraction: Optimal Transient Detection, Photometry, and Hypothesis Testing. ApJ 830, 27. doi:10.3847/0004-637X/830/1/27 [arXiv:1601.02655]

A. Glossary of features used

**SExtractor Features calculated from sepfreadthedocs.io**

- **thresh** (float) Threshold at object location.
- **npix** (int) Number of pixels belonging to the object.
- **tnpix** (int) Number of pixels above threshold (unconvolved data).
- **xmin, xmax** (int) Minimum, maximum x coordinates of pixels.
- **ymin, ymax** (int) Minimum, maximum y coordinates of pixels.
- **x, y** (float) object barycenter (first moments).
- **x2, y2, xy** (float) Second moments.
- **errx2, erry2, errxy** (float) Second moment errors.
- **a, b, theta** (float) Ellipse parameters, scaled as described by Section 8.4.2 in The Source Extractor Guide
- **cxx, cyy, cxy** (float) Alternative ellipse parameters.
- **clflux** (float) Sum of member pixels in convolved data.
- **flux** (float) Sum of member pixels in unconvolved data.
- **cpeak** (float) Peak value in convolved data.
- **peak** (float) Peak value in unconvolved data.
- **xcpeak, ycpeak** (int) Coordinate of convolved peak pixel.
- **xpeak, ypeak** (int) Coordinate of unconvolved peak pixel.
- **fflag** (int) Extraction flags.

**SEExtractor Features calculated from Holwerda (2005), and SExtractor’s documentation**

- **FLUX_ISO** Isophotal flux
- **FLUXERR_ISO** RMS error for isophotal flux
- **MAG_ISO** Isophotal magnitude
- **MAGERR_ISO** RMS error for isophotal magnitude
- **FLUX_AUTO** Flux within a Kron-like elliptical aperture
- **FLUXERR_AUTO** RMS error for AUTO flux
- **MAG_AUTO** Kron-like elliptical aperture magnitude
- **MAGERR_AUTO** RMS error for AUTO magnitude count
- **FLUX_APER** Flux vector within fixed circular aperture(s)
- **FLUXERR_APER** RMS error vector for aperture flux(es)
- **MAG_APER** Fixed aperture magnitude vector
- **MAGERR_APER** RMS error vector for fixed aperture
- **MU_THRESHOLD** Detection threshold above background
- **MU_MAX** Peak surface brightness above background
- **BACKGROUND** Background at centroid position
- **THRESHOLD** Detection threshold above background
- **XMIN_IMAGE Minimum x-coordinate among detected pixels
- **XMAX_IMAGE Maximum x-coordinate among detected pixels
- **YMIN_IMAGE** Minimum y-coordinate among detected pixels
- **YMAX_IMAGE** Maximum y-coordinate among detected pixels
- **YPEAK_IMAGE** y-coordinate of the brightest pixel
- **XPEAK_IMAGE** x-coordinate of the brightest pixel
- **X_IMAGE** Object position along x
- **Y_IMAGE** Object position along y
- $X^2_{\text{IMAGE}}$ Variance along x
- $Y^2_{\text{IMAGE}}$ Variance of position along y
- $X_Y_{\text{IMAGE}}$ Covariance of position between x and y
- $A_{\text{IMAGE}}$ Profile RMS along major axis
- $B_{\text{IMAGE}}$ Profile RMS along minor axis
- $\text{THETA}_{\text{IMAGE}}$ Position angle (CCW/x) counterclockwise
- $C_{XX_{\text{IMAGE}}}$ Cxx object ellipse parameter
- $C_{YY_{\text{IMAGE}}}$ Cyy object ellipse parameter
- $C_{XY_{\text{IMAGE}}}$ Cxy object ellipse parameter
- $\text{ELONGATION}_{\text{A_{\text{IMAGE}}/B_{\text{IMAGE}}}}$
- $\text{ELLIPTICITY}_{1 - B_{\text{IMAGE}}/A_{\text{IMAGE}}}$
- $\text{SNR}_{\text{WIN}}$ Gaussian-weighted SNR
- $\text{FLUX}_{\text{MAX}}$ Peak flux above background
- $\text{CLASS}_{\text{STAR}}$ S/G classifier output

Features derived.
- $\text{ROUNDNESS}_{A_{\text{IMAGE}}/B_{\text{IMAGE}}}$
- $\text{PEAK}_\text{CENTROID} = \sqrt{(X_{\text{PEAK_{IMAGE}}} - X_{\text{IMAGE}})^2 + (Y_{\text{PEAK_{IMAGE}}} - Y_{\text{IMAGE}})^2}$