MNRAS **000**, 1–?? (2019) Preprint 29 March 2019 Compiled using MNRAS LATEX style file v3.0

A reference dataset for astronomical transient event recognition I: lightcurves and tests on classical machine learning algorithms

Mauricio Neira[1], Catalina Gómez[2], Diego A. Gómez [1], Marcela Hernández Hoyos[1], Pablo Arbeláez[2], Jaime E. Forero-Romero[3]
[1] *Systems and Computing Engineering Department, Universidad de los Andes, Cra. 1 No. 18A-10, Bogotá, Colombia*
[2] *Departamento de Ingeniería Biomédica, Universidad de los Andes, Cra. 1 No. 18A-10, Bogotá, Colombia*
[3] *Departamento de Física, Universidad de los Andes, Cra. 1 No. 18A-10, Bogotá, Colombia*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We introduce ATRANCCATA (Annotated TRANsient Catalina CATAlog) an annotated dataset of 4869 transient and 16940 non-transient object lightcurves built from the Catalina Real Time Transient Survey. We provide access to this dataset as a plain text file to facilitate standarized quantitative comparison of astronomical transient event recognition algorithms. Some of the classes included in the dataset are: supernovae, cataclismic variables, active galactic nuclei, high proper motion stars, blazars and flares. As an example on how to use the dataset we experiment with multiple data pre-processing methods, feature selection techniques and classic machine learning algorithms (Support Vector Machines, Random Forests and Neural Networks). We assess quantitative performance in two classification tasks: binary (transient/non-transient) and eight-class classification. The best performing algorithm is a Random Forest Classifier for both classification experiments. The next release of ATRANCCATA will include images and benchmarks with deep learning models. All our code and data is available to the community at `https://github.com/MachineLearningUniandes/ATRANCCATA`.

Key words: methods: data analysis, statistical

1 INTRODUCTION

The study and detection of astronomical variable sources is expected to occur on unprecedented scales with the new generation of forthcoming multi-epoch and multi-band (synoptic) astronomical surveys. For instance, projects like the Large Synoptic Survey Telescope (LSST) (Ivezić et al. 2008; Jurić et al. 2015) are expected to generate exuberant daily data-streams near to 20 petabytes every night.

One of the main challenges that these datasets want to address is Real-Time Transient classification, i.e. flagging astrophysically relevant events whose luminosity varies in short duration relative in the timescale of the Universe, from minutes to several years. Transients include phenomena such as supernovae, novae, neutron stars, blazars, pulsars, cataclysmic variable stars (CV), gamma ray bursts (GRB) and active galaxy nuclei (AGN). The time-domain dependency of these objects is one of the reasons why they are hard to classify: their data is usually heterogeneous, unbalanced, sparse, unevenly sampled and with missing information. This has motived the use of Machine Learning (ML) algorithms to recognize and classify transient events.

There have been successful attempts to implement these algorithms using images as an input. For instance, data from the SkyMapper Supernova and Transient Survey and the High cadence Transient Survey (HiTS) have been used as inputs to automatic detection algorithms (Gieseke et al. 2017; Cabrera-Vives et al. 2017). Convolutional Neural Networks (CNN) have also achieved high accuracy in this binary classification task. Other studies have shown that artifacts can be detected using features extracted from raw images. Klencki et al. (2016) achieved reliable classification by transforming transient data from the OGLE-IV data-reduction pipeline and training it with machine learning algorithms such as Artificial Neural Networks, Support Vector Machines, Random Forests, Naive Bayes, K-Nearest Neighbors and Linear Discriminant Analysis. Similarly, Wright et al. (2015) used images from Pan-STARS1 Medium Deep Surveys, and du Buisson et al. (2015) processed single-epoch multi-band images from the SDSS supernova survey for the same purpose.

A complementary approach uses the lightcurves computed from the images to perform the classification task. For instance D'Isanto et al. (2016) used classical machine learning algorithms such as Random Forest, MultiLayer Perceptron and K-Nearest Neighbours lightcurves to classify transients from the Catalina Real Time Transient Survey; Lochner et al. (2016) used the same approach to find where supernovas from the Supernova Photometric Classification Challenge.

A crucial element in the developmet of any ML algorithm is the compilation of the training dataset. In astronomy this task has been facilitated by the publication of large databases from different observational projects. However, the publications that make use of these datasets still make extensive use of the internal know-how of the scientific collaboration. It is still difficult for another scientists to rebuild a training dataset and perform comparisons with published results.

To address this issue we compile and publish in easy-to-access files a dataset that can be used to train and test different ML algorithms for transient detection. We use public data from the Catalina Real-Time Transient Survey (CRTS) (Drake et al. 2012), an astronomical survey searching transient and highly variable objects as base for the dataset. In this paper we present lightcurve data, in a future publication we will present an imaging dataset.

In Section 2 we present the CRTS and the steps we follow to build the dataset. In Section 3 we describe its main features together with the repository structure gathering the files and some Python code to explore it. In Section 4 we show how this dataset can be used to perform some tests using some classical ML tests follow a similar approach as D'Isanto et al. (2016). We finalize in Section 5 with a summary of main features of our dataset and the results of our ML tests.

2 THE LIGHTCURVE DATASET

We use public data from the Catalina Real-Time Transient Survey (CRTS) (Drake et al. 2009), an astronomical survey searching transient and highly variable objects. CRTS covered 33000 squared degrees of sky and took data since 2007. Three telescopes were used: Mt. Lemmon Survey (MLS), Catalina Sky Survey (CSS), and Siding Spring Survey (SSS). So far, CRTS has discovered more than 15000 transient events. We use data from the CSS telescope, which is an f/1.8 Schmidt telescope located in the Santa Catalina Mountains in Arizona. The telescope is equipped with a 111-megapixel detector, and covered 4000 square degrees per night, with a limiting magnitude of 19.5 in the V band.

Putting together the lightcurves for ATRANCCATA implies cross-matching different files in the legacy CRTS webpage published as here: `http://nesssi.cacr.caltech.edu/DataRelease/CRTS-I_transients.html`. The photometry is stored in two different kind of files: `phot` that come from the main photometry database and `orphan` that correspond to transients not associated with the 500 million sources in the main photometry database. There are also `out` files that must be used to link transient IDs to database IDs.

For each one of the 5540 transients reported and classified in the archival webpage `http://nesssi.cacr.caltech.edu/catalina/All.arch.html` we use its transient IDs and its database IDs to look for the lightcurves in the `phot` and `orphan` files. Only 4982 transients can be linked to available data to reconstruct its lightcurves. Furthermore, some of these lightcurves are duplicated, i.e. they had the same number of observations, MJD and magnitude measurement. We drop the duplicates to end up with 4869 unique transients with an associated lightcurve. Figure 1 shows this process.

Class	Object Count
SN	1723
CV	988
HPM	640
AGN	446
SN?	319
Blazar	243
Unknown	228
Flare	219
AGN?	138
CV?	77

Table 1. Top 10 transient classes, with their respective number of lightcurves.

The CRTS dataset already provides a classification. The most numerous classes are: supernovae (SN), cataclysmic variable stars (CV), blazars, flares, asteroids, active galactic nuclei (AGN), and high-proper-motion stars (HPM). Though most objects in the transient object catalogue belong to a single class, there is some uncertainty in the categorization of some of them. In this case, an interrogation sign is used when a class is not clear e.g. SN? or sometimes multiple possible classes are found for a single event e.g. SN/CV. Table 1 summarizes the number of objects in each class.

To compile the non-transient lightcurves we retrieve sources in the dataset from the CRTS online catalogue by retrieving objects within a 20 arcsecond radius from CRTS detected transients, and removing known transient lightcurves from that set. We end up with 16940 unique non-transient lightcurves. Figure 2 shows this process.

Figure 3 shows the number of lightcurves as a function of average magnitude (left panel) and as a function of the number of points in the lightcurve. We show separately the whole data set and three representative classes: supernova (SN), cataclismic variables (CV) and active galactic nuclei (AGN). For these four sets the median magnitude is in the range $18 - 20$. The number of points in the lightcurve has a larger variability. The median for all the curves is close to 30, while for SN, CV and AGN it is close to 15, 50 and 180, respectively.

3 REPOSITORY DESCRIPTION

The repository that contains the lightcurves and a jupyter notebook to reproduce some of the Figures and Tables in this paper. The repository can be found in `https://github.com/MachineLearningUniandes/ATRANCCATA`. To date the repository has three main folders.

- `data/lightcurves`: contains three text

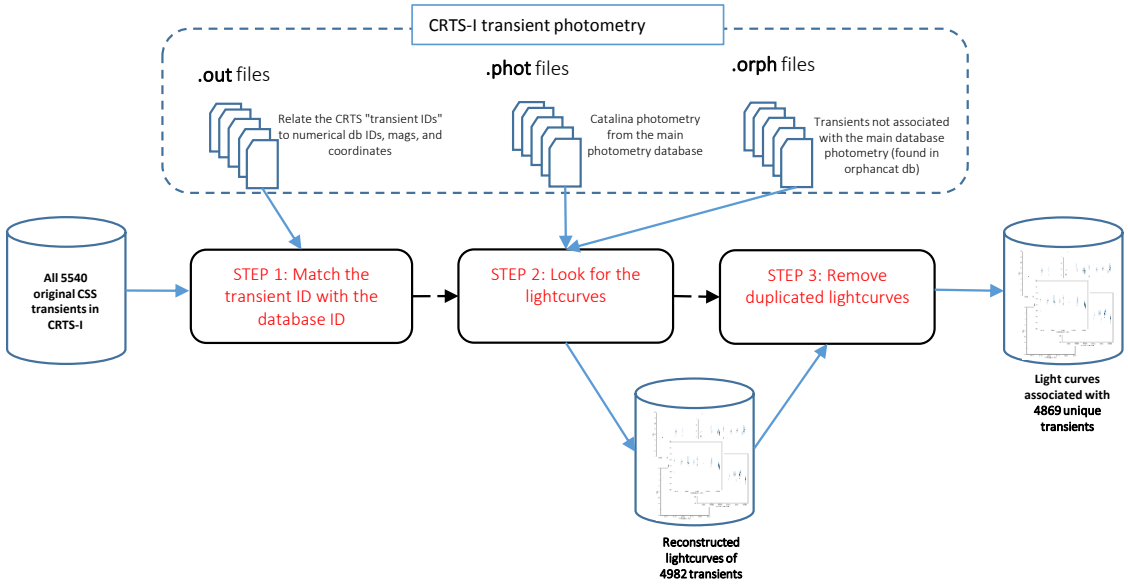

Figure 1. ATRANCCATA Dataset Set Up: Lightcurve compilation for transients.

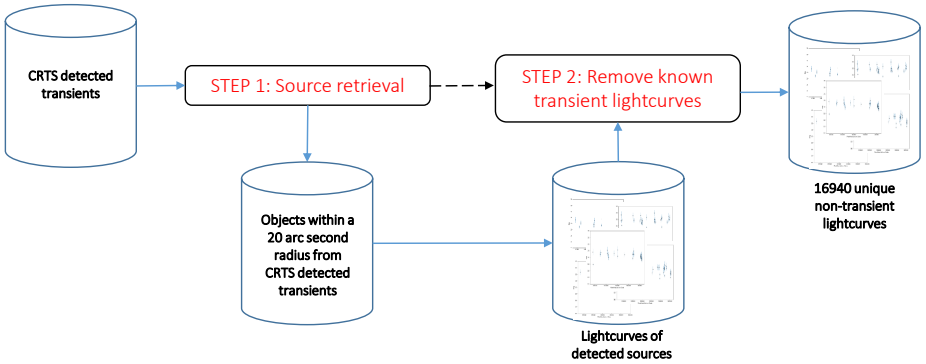

Figure 2. ATRANCCATA Dataset Set Up: Lightcurve compilation for non-transients.

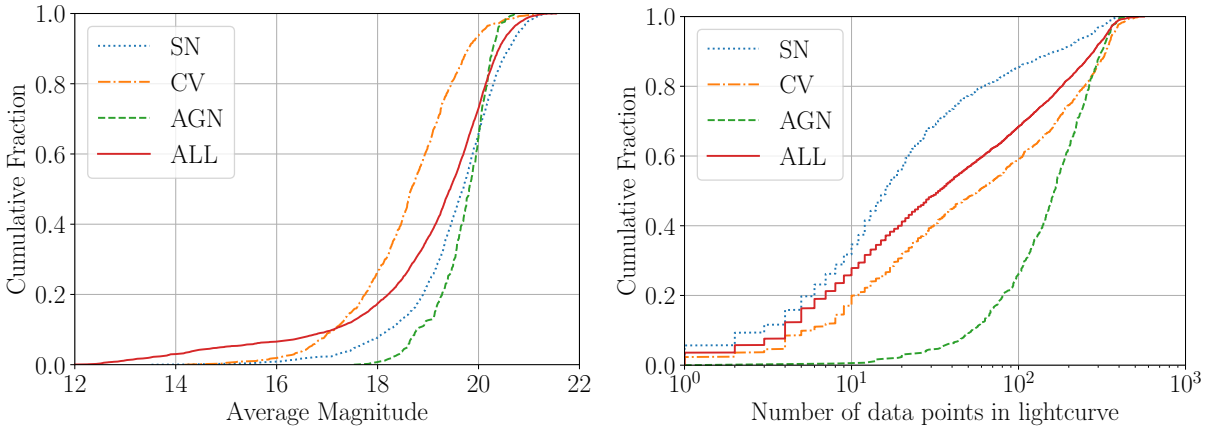

Figure 3. Cumulative number of lightcurves (expressed as a fraction) as a function of average magnitude (left) and number of data points in the lightcurve (right). This includes information for the three most representative classes (SN, CV, AGN) and the whole database (ALL).

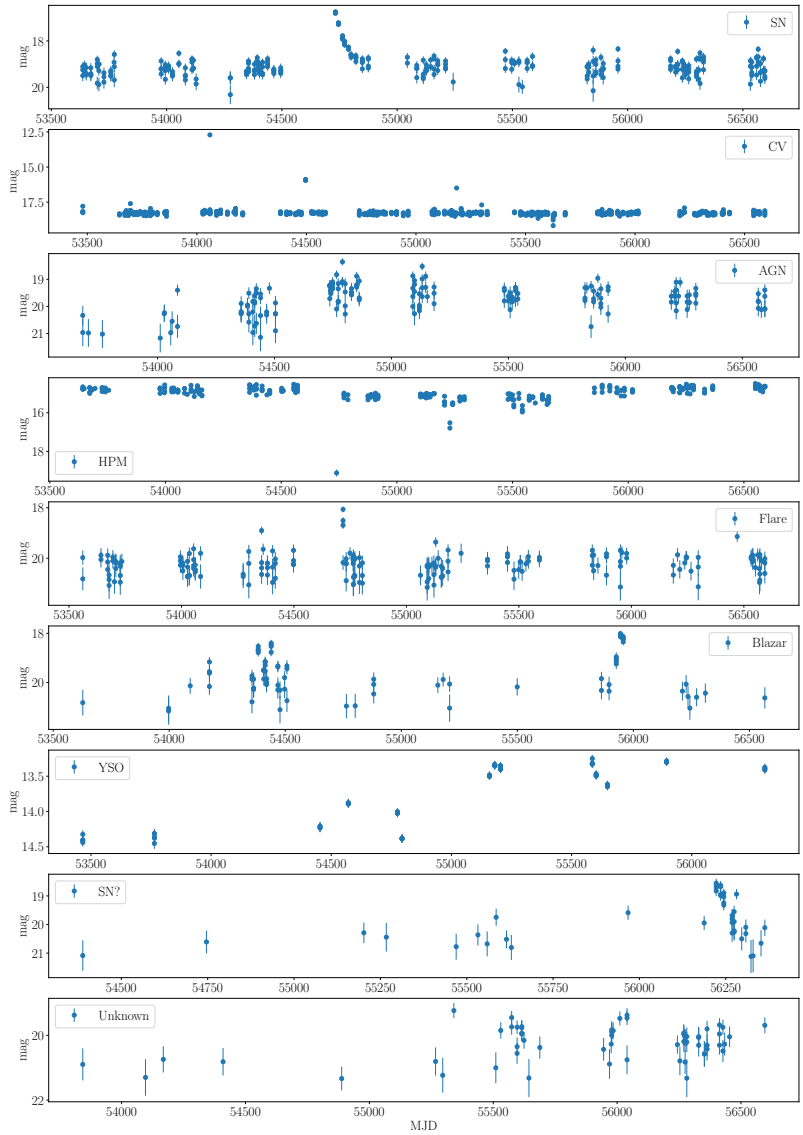

Figure 4. Randomly selected lightcurves for the most represented classes as compiled in the database we present in this paper.

files in CSV format the transient lightcurves (`transient_lightcurves.csv`), the labels for the transients (`transient_labels.csv`) and the lightcurves for non-transient objects (`nontransient_lightcurves.csv`). The first two files can be linked by unique transient IDs and provided in the CRTS database.

• `nb-explore`: includes a jupyter notebook (`explore_light_curves.ipynb`) with examples on how to read and plot transient and non-transient lightcurves (Figure 4 and Figure 5), extract the statistic in Table 1 and prepare the summary statistics in Figure 3. There is some additional python files (`features.py`, `helpers.py` and `inputs.py`) to read and perform simple operations on the CSV data files.

• `papers/lightcurves`: includes the source `tex` files and figures for this paper.

4 CLASSICAL MACHINE LEARNING TESTS

As an example on how the dataset can be used, we apply classical machine learning (ML) algorithms to perform different classification tasks (Figure 6).

4.1 Features

We do not feed directly the anotated lightcurves to the ML algorithms. We perform some preprocessing as follows. First, we discard lightcurves with less than 10 data points observations as they may not contain enough information to be classified correctly.

Given that the number of lightcurves per class is unbalanced, in order to have the same amount elements for each class we implement an oversampling step by artificially generating multiple mock lightcurves, each based on an observed one. We generate a mock lightcurve from the observed lightcurve and then sample the observed magnitude from a

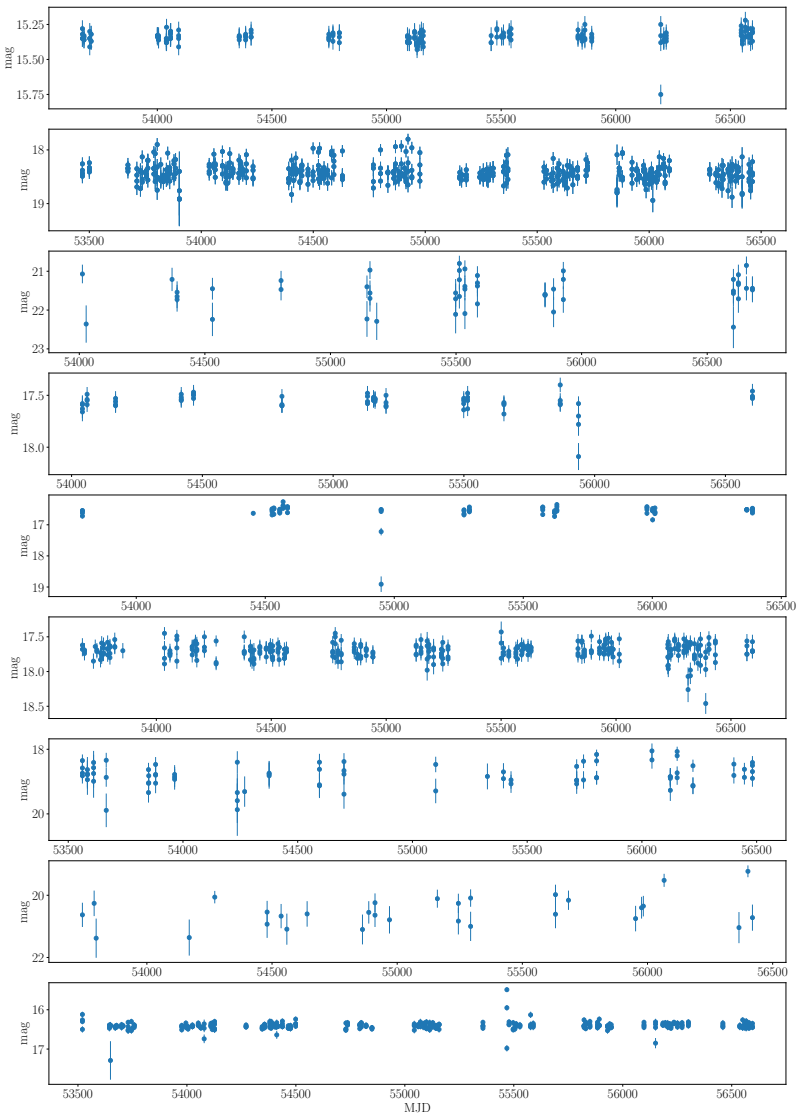

Figure 5. Randomly selected lightcurves for non-transient sources.

Gaussian distribution centered on the observational apparent magnitude with the magnitude's error as the standard deviation.

Finally, we compute a fixed set of features for each lightcurve. These features are scalars derive from statistical and model-specific fitting techniques. The first features (moment-based, magnitude-based and percentile-based) were formally introduced in Richards et al. (2011), and have been used in other studies (Lochner et al. 2016; D'Isanto et al. 2016). We extend that list to include another set (polynomial fitting-based features). At the end of this process we renormalize the features to have zero mean and unit variance.

These groups of features are:

(i) Moment-based features, which use the magnitude for each lightcurve.

- `beyond1std`: Percentage of observations which are over or under one standard deviation from the weighted average. Each weight is calculated as the inverse of the corresponding observation's photometric error.
- `kurtosis`: The fourth moment of the data distribution.
- `skew`: Skewness. Third moment of the data distribution.
- `sk`: Small sample kurtosis.
- `std`:: The standard deviation.
- `stetson_j`: The Welch-Stetson J variability index (Stetson 1996). A robust standard deviation.
- `stetson_k`: The Welch-Stetson K variability index (Stetson 1996). A robust kurtosis measure.

(ii) Features based on the magnitudes.

- `amp`: The difference between the maximum and minimum magnitudes.
- `max_slope`: Maximum absolute slope between two consecutive observations.
- `mad`: The median of the difference between magnitudes and the median magnitude.

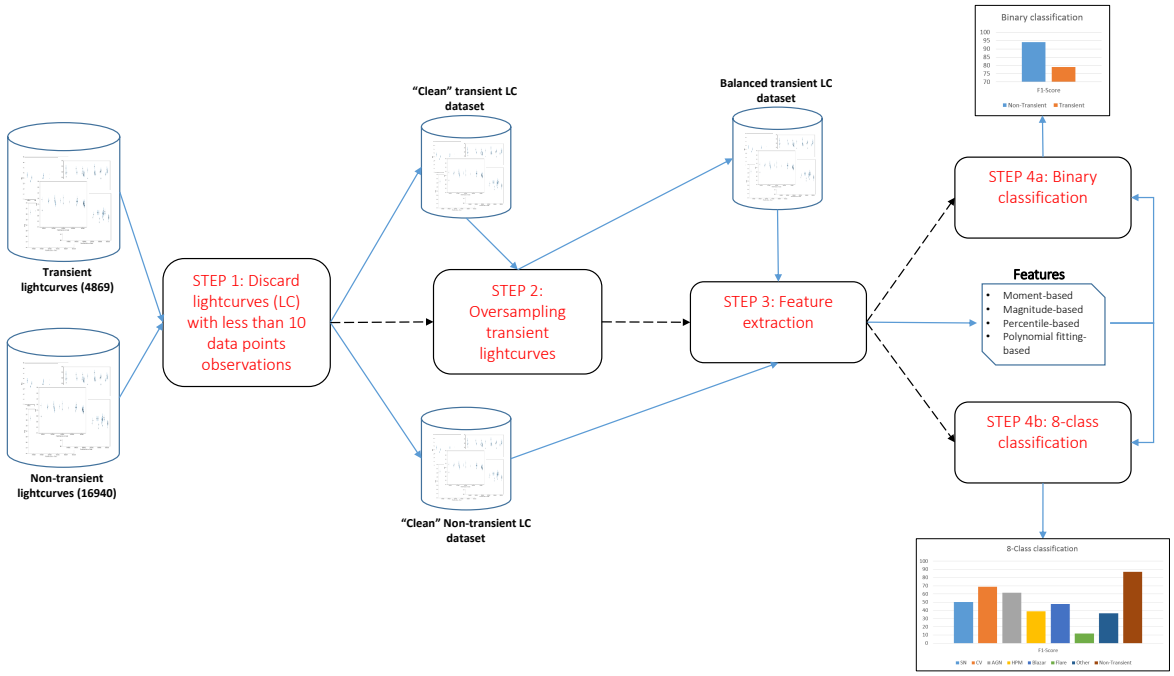

Figure 6. ATRANCCATA Machine Learning process.

- `mbrp`: The percentage of points within 10% of the median magnitude.
- `pst`: Percentage of all pairs of consecutive magnitude measurements that have positive slope.
- `pst_last30`: Percentage of the last 30 pairs of consecutive magnitudes that have a positive slope, minus percentage of the last 30 pairs of consecutive magnitudes with a negative slope.

(iii) Percentile-based features, which use the sorted flux distribution for each source. The flux is computed as $F = 10^{0.4\mathrm{mag}}$. We define $F_{n,m}$ as the difference between the m-th and n-the flux percentiles.

- `p_amp`: Largest percentage difference between the absolute maximum magnitude and the median.
- `pdfp`: Ratio between $F_{5,95}$ and the median flux.
- `fpr20`: Ratio $F_{40,60}/F_{5,95}$
- `fpr35`: Ratio $F_{32.5,67.5}/F_{5,95}$
- `fpr50`: Ratio $F_{25,75}/F_{5,95}$
- `fpr65`: Ratio $F_{17.5,82.5}/F_{5,95}$
- `fpr80`: Ratio $F_{10,90}/F_{5,95}$

(iv) Polynomial Fitting-based features, which are the coefficients of multi-level terms in a polynomial curve fittingpr. This is a new set of features proposed in this paper. `Polyn_Tm` indicates the coffecient of the term of order `m` in a fit to a polinomial of order `n`.

- `Poly1_T1`.
- `Poly2_T1`.
- `Poly2_T2`.
- `Poly3_T1`.
- `Poly3_T2`.
- `Poly3_T3`.
- `Poly4_T1`.
- `Poly4_T2`.
- `Poly4_T3`.
- `Poly4_T4`.

4.2 Classification Tasks

We study two classification tasks.

- Binary Classification. Using a balanced number of events from both classes in order to investigate the capability of distinguishing between Transients and Non-Transients.
- 8-Class Classification. Using the unbalanced number of objects across classes to perform a classification into the following categories: AGN, Blazar, CV, Flare, HPM, Other, Non-Transient and Supernovae.

4.3 ML algorithms

We conduct experiments with three widely used families of supervised classification algorithms: Neural Networks (NNs), Random Forests (RFs) and Support Vector Machines (SVMs).

These algorithms are popular in published studies and are efficient for low dimensional feature datasets as is our case. We use sklearn (Pedregosa et al. 2011) Python's implementation of these algorithms. Details on the inner workings of these machine learning models can be found in Hastie et al. (2016). The set of hyperparameter space explored for each algorithm is the following.

- Neural Networks:
  - Learning Rate: Either constant vs. adaptive.
  - Hidden Layer Sizes: Single Layer with 100 nodes vs. Two layers with 100 nodes each.
  - L2 Penalty (α): 10^{-1}, 10^{-2}, 10^{-3} , 10^{-4}.
  - Activation Function: Logistic vs. Relu.

	Precision	Recall	f1-Score	Support
Non-Transient	94.13	94.13	94.13	3798
Transient	79.10	79.10	79.10	1067

Table 2. Precision, Recall and f1-score for the Binary Classification Task with Regular inputs.

	non-transient	transient
non-transient	3575	223
transient	223	844

Table 3. Confusion Matrix for the best performing model in the Binary task.

- Random Forest:
  - Number of Estimators: 200 or 700.
  - Number of features considered: Square Root or the Logarithm base 2 of the total number of features.
- Support Vector Machines:
  - Kernel: Radial Basis Function (RBF).
  - Kernel Coefficient (γ): 10^{-1}, 10^{-2}, 10^{-3} , 10^{-4}, 10^{-5}
  - Error Penalty (C): 1 vs 10 vs 100 vs 1000.

4.4 Validation

We split the input lightcurves in a training and test datasets. The test dataset contains 4869 unique transients lightcurves and , without any oversampling. We use 2-fold cross-validation during training as evaluation protocol. Moreover, we use grid search during training to test multiple hyper-parameter configurations for each one of the possible algorithms. We use the F1-Score to assess the performance of a given model and we evaluate each task on the held-out test dataset.

4.5 Results

4.5.1 Binary Classification

The best algorithm in this task is RFs with a maximum F1-score of 87.69%. SVMs are the second best-performing model with the F1-score of 85.36%. Changing the number of features does not affect significantly the score. NNs are ranked third, although their scores are very similar to those of SVMs. The highest achieved score for NNs is 85.03%.

Table 3 shows the confusion matrix of the best performing algorithm and Table 2 summarizes the scores. These results imply that non-transients are better classified overall.

Figure 7 displays the most important features for the RFs classifier. The top five inputs for classification are `stetson_j`, `std`, `mad`, `poly1_t1` and `poly2_t1`. The first feature achieved the highest importance of 21%, compared to the following with values in the range 6% - 8%.

	Precision	Recall	f1-Score	Support
SN	48.82	51.39	50.07	323
CV	66.96	70.69	68.77	215
AGN	48.14	85.84	61.69	106
HPM	25.19	86.84	39.05	76
Blazar (Bl)	46.77	49.15	47.93	59
Flare (Fl)	7.00	41.17	11.96	51
Other (O)	31.11	44.01	36.46	234
Non-Transient (NT)	96.06	79.69	87.12	3798
avg/total	46.25	63.59	50.38	4862

Table 4. Precision, Recall and f1-score for the 8-Class Classification Task with Regular inputs.

4.5.2 Eight-Class Classification

For this task RFs are againt the best classifier. The best f1-score is 66.05%. NNs are the second best. Its highest f1-score is 60.19%, while SVMs are the worst-performing model only achieving a maximum f1-score of 57.30%. Table 4 summarizes the results and Table 5 presents the confusion matrix for the RF.

The two classes with highest recall are HPM and Non-Transient, with a recall of 86.36% and 84.13%, respectively. The worst performing classes are Blazar, Flare and Other, with recall values in the range 36% - 40%. SN is the class with which most other class instances are incorrectly classified. Moreover, Flares have about 50% of the test samples classified as Non-Transients, AGNs have about 20% of their samples classified as Other, and Blazars and Other had most of its samples classified as AGN. Additionally, most incorrectly classified AGNs (∼20.5%) are identified as Other and most Blazar instances are incorrectly categorized as either SN or AGN.

Figure 8 displays the feature importance ranking. This list ranks first `stetson_j` with an 8% importance, followed by `amp`, `sk`, `std`, `mad`, with values around 6%.

5 CONCLUSIONS

The scope of forthcoming of large astronomical synoptic surveys motivates the development and exploration of automatized ways to detect transient sources. In turn this prompts the compilation of publicly available databases to train and test new algorithms. In this paper we presented the results of such compilation based on data from the Catalina Real-Time Transient Suvey (CRTS). The data-set compiles 4869 transient and 16940 non-transient lightcurves. The dataset is publicly available at `https://github.com/MachineLearningUniandes/ATRANCCATA`.

We illustrated how to use this database by extracting characteristic features to use them as inputs to train three different machine learning algorithms (Random Forests, Neural Networks and Support Vector Machines) for classification tasks. The features extracted from lightcurves were ei-

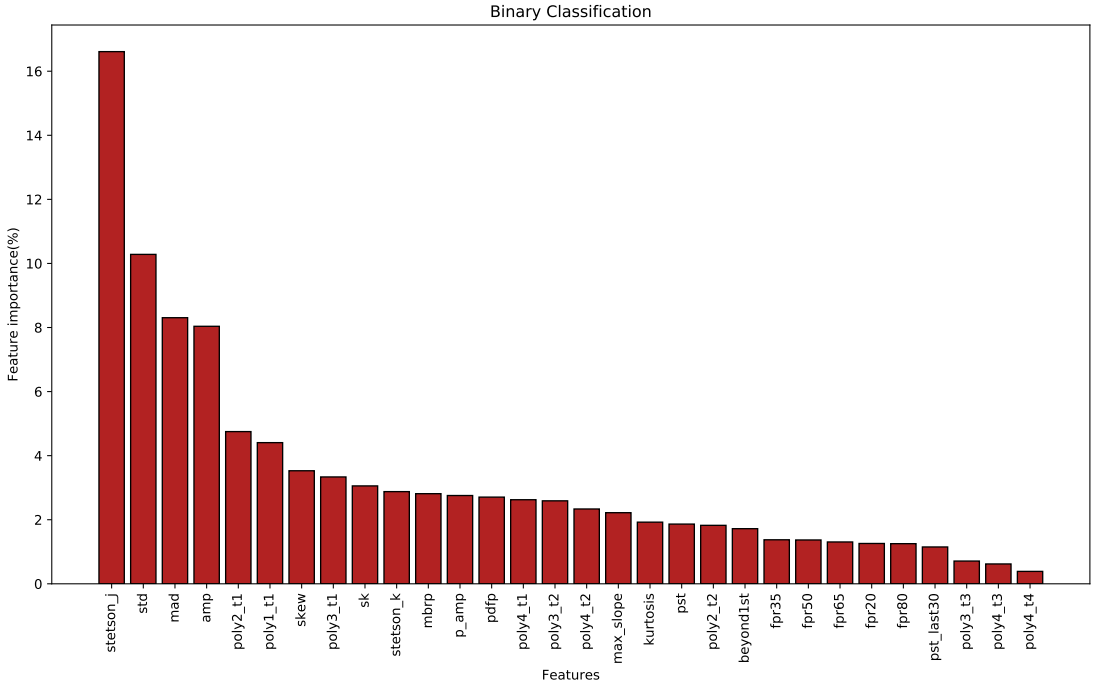

Figure 7. Feature importance rank for the best Random Forest classifier for the Binary classification task. Feature importance is represented with percentages.

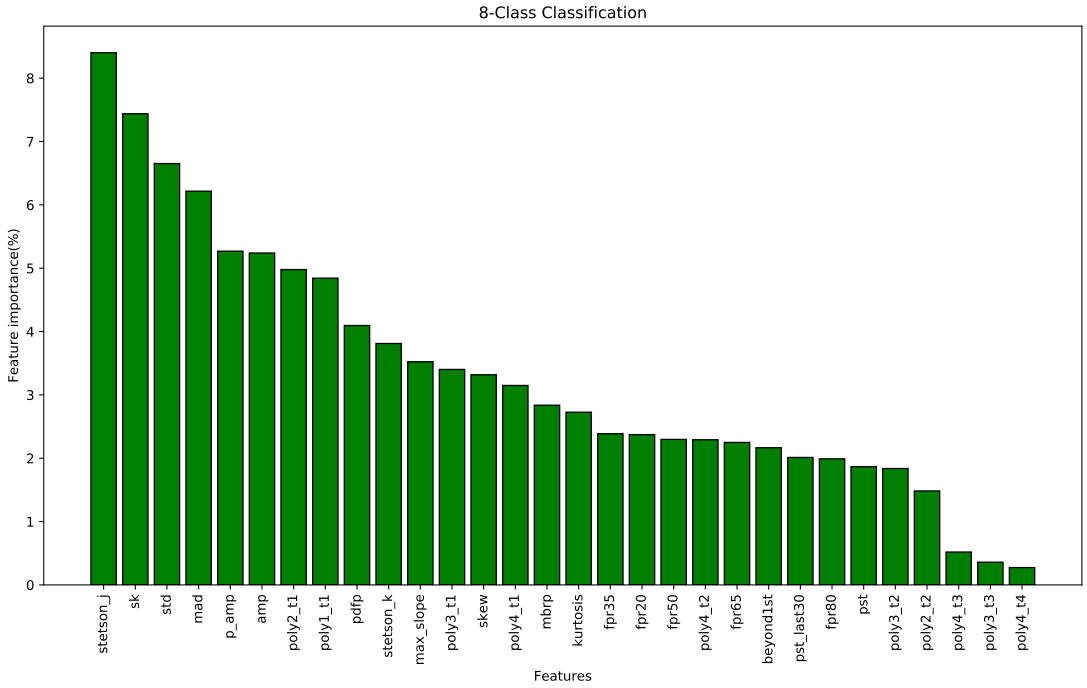

Figure 8. Feature importance rank for the best Random Forest classifier for the best 8-Class classification task. Feature importance is represented with percentages.

	SN	CV	AGN	HPM	Blazar	Flare	Other	Non-Transient
SN	166	25	0	0	7	5	40	97
CV	17	152	0	1	5	3	12	37
AGN	1	2	91	0	10	1	35	49
HPM	5	0	0	66	0	0	5	186
Blazar	8	13	4	0	29	0	6	2
Flare	16	5	0	0	3	21	4	251
Other	53	12	7	1	3	3	103	149
Non-Tr.	57	6	4	8	2	18	29	3027

Table 5. Confusion Matrix for the best performing model in the 8-Class task. The classes follow the abbreviations in Table 4

ther statistical descriptors of the observations, or polynomial curve fitting coefficients applied to the lightcurves. Overall the best classifier for all tasks was the Random Forest. In this model the most important feature was always `stetson_j`, i.e. a robust estimate for the standard deviation.

In a second paper we will present another reference dataset for astronomical transient event recognition based on images of the CRTS. The corresponding tests will use state-of-the art deep learning techniques for transient classification.

ACKNOWLEDGEMENTS

We thank Andrew Drake for sharing with us the CRTS Transient dataset used in this project. We thank Juan Pablo Reyes, Dominique Fouchez for helping with the research. We acknowledge funding from Universidad de los Andes in the call for project finalization. We also thank contributors and collaborators of the SciKit-Learn, Jupiter Notebooks and Pandas' Python libraries.

CRTS and CSDR2 are supported by the U.S. National Science Foundation under grant NSF grants AST-1313422, AST-1413600, and AST-1518308. The CSS survey is funded by the National Aeronautics and Space Administration under Grant No. NNG05GF22G issued through the Science Mission Directorate Near-Earth Objects Observations Program.

REFERENCES

Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J.-C., 2017, ApJ, 836, 97

D'Isanto A., Cavuoti S., Brescia M., Donalek C., Longo G., Riccio G., Djorgovski S. G., 2016, MNRAS, 457, 3119

Drake A. J., et al., 2009, ApJ, 696, 870

Drake A. J., et al., 2012, in Griffin E., Hanisch R., Seaman R., eds, IAU Symposium Vol. 285, New Horizons in Time Domain Astronomy. pp 306–308 (arXiv:1111.2566), doi:10.1017/S1743921312000889

Gieseke F., et al., 2017, MNRAS, 472, 3101

Hastie T., Tibshirani R., Friedman J., 2016, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics). Springer

Ivezić Ž., et al., 2008, preprint, (arXiv:0805.2366)

Jurić M., et al., 2015, preprint, (arXiv:1512.07914)

Klencki J., Wyrzykowski Ł., Kostrzewa-Rutkowska Z., Udalski A., 2016, Acta Astron., 66, 15

Lochner M., McEwen J. D., Peiris H. V., Lahav O., Winter M. K., 2016, ApJS, 225, 31

Pedregosa F., et al., 2011, Journal of machine learning research, 12, 2825

Richards J. W., et al., 2011, ApJ, 733, 10

Stetson P. B., 1996, pasp, 108, 851

Wright D. E., et al., 2015, MNRAS, 449, 451

du Buisson L., Sivanandam N., Bassett B. A., Smith M., 2015, MNRAS, 454, 2026