MNRAS **000**, 1–?? (2018) Preprint 8 November 2018 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

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

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

The arrival of massive multi-epoch and multi-band astronomical surveys demands the development of computational techniques to automate the study and detection of transient astronomical sources. In this paper we put together an annotated dataset of more than 10000 transient and non-transient object light-curves from the Catalina Real Time Transient Survey (CRTS). This dataset provides a baseline to facilitate standarized quantitative comparison of astronomical transient event recognition algorithms. The classes included in the dataset are: supernovae, cataclismic variable, 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. We assess quantitative performance in five classification tasks, paying particular attention to a binary and a four multi-class classifications tasks. The best performing algorithm is a Random Forest Classifier across all these classification experiments. The next release of this reference database will include images and benchmarks with deep learning models.

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, the Large Synoptic Survey Telescope (LSST) (Ivezić et al. 2008), one of the largest synoptic survey telescopes to come in the following years, will generate approximately 15 terabytes of data every night (Jurić et al. 2015). Other surveys including the Square Kilometer Array (SKA) are also expected to generate exuberant daily data-streams.

This observational leap renders manual classification techniques unfeasible. Traditionally, such objects have been classified by visual inspection by experts or through crowd-sourcing (Smith et al. 2011; Sako et al. 2008). This approach is slow and expensive. Another concern is the possible biases and difficulty to standardize among astronomers (Bloom & Richards 2012). Alternatively, transient detection can be executed much faster than human astronomers through computational techniques using machine learning, which are deterministic and calculate the results' degrees of certainty. These methods also allow for real-time triggering of follow-up observations that optimize the economical and temporal resources.

Another challenge in Time Domain Astronomy is Real-Time Transient classification. Astronomical Transients are 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. Automating the recognition and classification of transient events, a type of such variable sources, would reduce costs and speed up this pro-

★ E-mail: da.gomez11@uniandes.edu.co

cess as well as provide scientists information of the universe in various spatial scales (Graham et al. 2012).

Transient detection is generally done through difference imaging (Kessler et al. 2015; Masci et al. 2017; Jian et al. 2017). This algorithm starts by aligning the image of interest on a reference image of the same region of sky, processing the former to match its point-spread function in all regions with the latter's, and subtracting both (Alard & Lupton 1998). As a result, variable stars and transient objects which were not visible in the reference image will remain in the resulting image. Nonetheless, the difference images can also contain artifacts due to imperfections in the image processing phase, in the telescope or other phenomena. Unfortunately, distinguishing between bogus and real transient objects still requires human intervention, making it expensive and slow (Graham et al. 2012).

There have been successful attempts to implement automatic detection algorithms and distinguish artifacts from real transient objects. For instance, raw images 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.

Some alternative approaches classify astronomical transient events without image subtraction. This is usually done by extracting relevant features (e.g. periodic, non-periodic) from the astronomical object light-curves, and using machine learning algorithms for classification. For instance, in Faraway et al. (2016) and D'Isanto et al. (2016), light curves of objects six or more transient classes from the Catalina Real Time Transient Survey and the Downes set (Downes et al. (2005)) were classified using this approach. Lochner et al. (2016) used the same approach to find where supernovas from the Supernova Photometric Classification Challenge. Recurrent Neural Networks (RNNs) have also been proven successful to classify transient events, Charnock & Moss (2017) and Hinners et al. (2018) showed that modern algorithms can also learn from time-series data without expensive image processing.

This project is a compiled study that unifies techniques used in previous studies, in a structured way. We openly provide the curated datasets and the code used in this project, so that others may replicate the results and expand upon them (Section 3). Thus, with this paper we provide an experimental framework that can be used as a baseline for future research on automated transient object recognition.

In this paper we follow the mentioned approach and test different machine learning algorithms to classify transient events using light curves as an input. We start by extracting several statistical descriptors from the light curves. These descriptors are the input of three different machine learning models with several hyper-parameter variations: Support Vector Machines (SVMs), Neural Networks (NNs) and Random Forests (RFs), which automatically learn to detect and classify between different types of transient objects. We present the results of extensive testing and multiple experiments executed in order to find the best training parameters.

The paper is structured as it follows. In Section 2 we present the dataset used for this project. Section 4 describes the methodology. Section **??** explains the experiments performed to classify transient objects with machine learning. Finally, the results are presented and discussed in Section **??**.

2 DATA

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. It 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 light curves as measured with the CSS telescope of the CRTS, which is an f/1.8 Schmidt telescope located in the Santa Catalina Mountains, north of Tucson, Arizona and 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. The public CRTS data base reports the source flux and its corresponding uncertainty (Stetson 1996).

All transient objects were classified in the CRTS dataset according to their type. The most relevant 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 summarized the number of objects in each class.

We use the light curves of 4269 unique transient events that contain at least 5 observations. We also use 15193 non-transient sources with at least 5 observations each. We obtained sources from the transient dataset directly from researchers of the CRTS project. Alternatively, we retrieved sources in the dataset from the CRTS online catalogue, by sampling light curves of objects within a 0.006 degree radius from CRTS detected transients, and removing known transient light curves from that set. Though this process should return only non-transient sources, it is possible that non-detected transients were captured and catalogued incorrectly as 'non-transients'. Table 2 and Table 3 summarize some statistics on the number of observations available for each light curve for the transient and non-transient datasets, respectively.

Class	Object Count
SN	1293
CV	862
AGN	425
HPM	306
Blazar	237
SN?	236
Flare	207
AGN?	130
Unknown	114
CV?	55

Table 1. Top 10 transient classes, with their respective event count.

3 REPOSITORY DESCRIPTION

The repository that contains the code, data and trained models we used in this project is found in the website https://github.com/diegoalejogm/crts-transient-recognition. For this project we used various python libraries: `jupyter` (1.0.0), `numpy` (1.14.3), `pandas` (0.22.0) and `scikit-learn` (0.19.1). Details other dependencies can be found in the included file `requirements.txt`.

We organized the repository in three main folders. The first one is `data`, and it contains the data we used in this project. We split data in three directories: `lightcurves`, `features` and `inputs`. The former contains pandas dataframes for raw and curated light-curves. The `features` subfolder contains the resulting features of preprocessing the light-curves, for both transients and non-transients. Finally, the `inputs` subfolder contains the already split training and testing inputs for the following hyper-parameter combinations: classification task, number of features, minimum number of observations, and balanced/unbalanced data. Each one of the input files contains a numpy tuple with the structure: *(train features, train labels, test features, test labels)*

The second main folder in the repository is `notebooks`. It contains all the code used for this project. The jupyter notebooks contain the pipeline of our proposed experimental framework, and they are numbered in sequential order. Moreover, each notebook is documented within itself. Two additional non-numbered notebooks are used for exploration purposes. Conversely, the `.py` files are used to contain the python code which is used in the notebooks. They're separated by task type.

Finally, the `results` directory contains the products of running the code. One result type is named dataframes, which contains a pandas dataframe for each task. Each dataframe contains the testing and training results, including scores and confusion matrices. On the other hand, the results sub-folder contains the feature importance list figures presented in this document.

For more information on the repository, make sure to read the README file included, or email us directly.

The tables 2 and 3 in this annex contain statistical descriptors of the light curves' observation count, for transient and non-transient curves that contain at least 5 observations. Table 2 makes reference to transient events, whereas table 3 refers to non-transient objects. Each of these tables present the mean, standard deviation, percentiles (25, 50 and 75) and maximum number of observations per light curve found in each dataset.

Count	4269
Mean	102.810
Std	113.786
Min	5
25%	14
50%	48
75%	166
Max	564

Table 2. Transient's data-set observation count per object.

Count	15193
Mean	118.37
Std	116.51
Min	5
25%	26
50%	72
75%	185
Max	537

Table 3. Non-Transient's data-set observation count per object.

4 CLASSICAL MACHINE LEARNING TESTS

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

4.1 Data Preprocessing

We do not feed directly the anotated lightcurves to the ML algorithms. There is preprocessing stage that follows six steps.

4.1.1 Data Filtering

We discard light curves with few observations as they may not contain enough information to be classified correctly Our nominal cut is 10 observations per light curve.

4.1.2 Oversampling Transient Light Curves

The number of light curves 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 light curves, each based on an observed one.

We generate a mock light curve from the observed light curve and then sample the observed magnitude from a Gaussian distribution centered on the observational apparent magnitude with the magnitude's error as the standard deviation.

4.1.3 Feature Extraction

Light curves are sampled at irregular time intervals and have different numbers of data points. Thus, it is challenging to directly use the time-series data for classification with traditional methods. We circumvent this problem by extracting a set of features for each light curve. 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. These groups are:

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

- Beyond1std (*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 (*kurtosis*): The fourth moment of the data distribution. Used to measure the heaviness or lightness in the tails of the statistical data.
- Skewness (*skew*): A measurement of the level of asymmetry from the normal distribution in a data distribution. Negative skewness is the property of a more pronounced left tail, while positive skewness is a characteristic that implies a more pronounced right tail.
- Small Kurtosis (*sk*): Small sample kurtosis.
- Standard deviation (*std*): The standard deviation of the magnitudes.
- Stetson J (*stetson_j*): The Welch-Stetson J variability index Stetson (1996). A robust standard deviation.
- Stetson K (*stetson_k*): The Welch-Stetson K variability index Stetson (1996). A robust kurtosis measure.

(ii) Magnitude-based features, which rely on the magnitude for each source.

- Amplitude (*amp*): The difference between the maximum and minimum magnitudes.
- Max Slope (*max_slope*): Maximum absolute slope between two consecutive observations.
- Median Absolute Deviation (*mad*): The median of the difference between magnitudes and the median magnitude.
- Median Buffer Range Percentage (*mbrp*): The percentage of points within 10% of the median magnitude.
- Pair Slope Trend (*pst*): Percentage of all pairs of consecutive magnitude measurements that have positive slope.
- Pair Slope Trend 30 (*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 $F = 10^{0.4\text{mag}}$. We define $F_{n,m}$ as the difference between the m-th and n-the flux percentiles.

- Percent Amplitude (*p _amp*): Largest percentage difference between the absolute maximum magnitude and the median.
- Percent Difference Flux Percentile (*pdfp*): Ratio between $F_{5,95}$ and the median flux.
- Flux Percentile Ratio Mid20 (*fpr20*): Ratio $F_{40,60}/F_{5,95}$
- Flux Percentile Ratio Mid35 (*fpr35*): Ratio $F_{32.5,67.5}/F_{5,95}$
- Flux Percentile Ratio Mid50 (*fpr50*): Ratio $F_{25,75}/F_{5,95}$
- Flux Percentile Ratio Mid65 (*fpr65*): Ratio $F_{17.5,82.5}/F_{5,95}$
- Flux Percentile Ratio Mid80 (*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 fitting. This is new set of features proposed in this paper.

- Poly1 T1: Linear term coeff. in monomial curve fitting.
- Poly2 T1: Linear term coeff. in quadratic curve fitting.
- Poly2 T2: Quadratic term coeff. in quadratic curve fitting.
- Poly3 T1: Linear term coeff. in cubic curve fitting.
- Poly3 T2: Quadratic term coeff. in cubic curve fitting.
- Poly3 T3: Cubic term coeff. in cubic curve fitting.
- Poly4 T1: Linear term coeff. in quartic curve fitting.
- Poly4 T2: Quadratic term coeff. in quartic curve fitting.
- Poly4 T3: Cubic term coeff. in quartic curve fitting.
- Poly4 T4: Quartic term coeff. in quartic curve fitting.

4.1.4 Feature Scaling

We re-scale the magnitudes to have zero mean and unit variance.

4.2 Classification Tasks

We study two classification tasks.

4.2.1 Binary Classification

We use a balanced number of events from both classes in order to investigate the capability of distinguishing between Transients and Non-Transients.

4.2.2 8-Class Classification

We consider 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.

For 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.

For 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.

For 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 light curves in a training and test datasets. The test dataset contains only original light curves, 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 5 shows the confusion matrix of the best performing algorithm and Table 4 summarizes the scores. These results imply that non-transients are better classified overall.

Figure 1 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 former achieved the highest importance with over 21%, compared to the following with values in the range 6% - 8%.

4.5.2 Eight-Class Classification

RFs are 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 **??** summarizes the results.

	Precision	Recall	f1-Score	Support
Non-Transient	85.41	90.94	88.09	1281
Transient	90.32	84.47	87.29	1281

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

	non-transient	transient
non-transient	90.94	9.06
transient	15.53	84.47

Table 5. Confusion Matrix for the best performing model in the Binary task.

	Precision	Recall	f1-Score	Support
AGN	65.62	66.14	65.88	127
Blazar	62.79	38.57	47.79	70
CV	77.53	74.89	76.19	235
Flare	66.67	39.29	49.44	56
HPM	90.48	86.36	88.37	88
Non-Transient	67.77	84.13	75.07	315
Other	47.02	36.60	41.16	194
SN	62.97	68.57	65.65	315
avg/total	66.39	66.93	66.05	1400

Table 6. Precision, Recall and f1-score for the 8-Class Classification Task with Regular inputs.

Table 7 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 ($\sim$20.5%) are identified as Other and most Blazar instances are incorrectly categorized as either SN or AGN.

Figure 2 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%. The lowest raking features are the five high level polynomial: poly4_t1, poly4_t2, poly3_t3, poly4_t3 and poly4_t4.

5 CONCLUSIONS

The scope of forthcoming of large astronomical synoptic surveys such as the LSST (Ivezić et al. 2008) motivates the

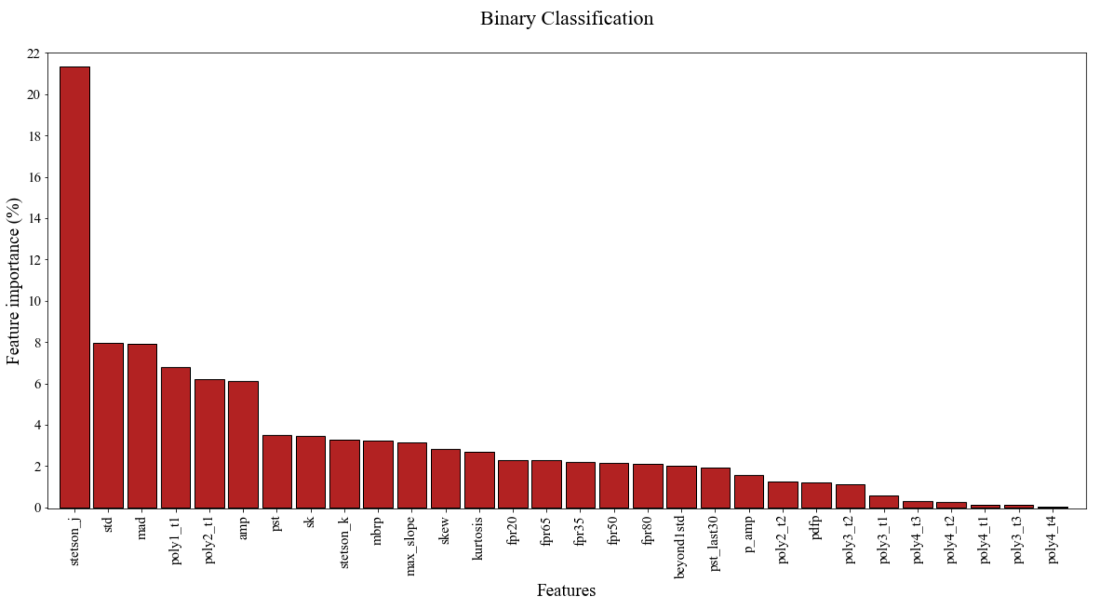

Figure 1. Feature importance rank for the best Random Forest classifier for the Binary classification task. Feature importance is represented with percentages.

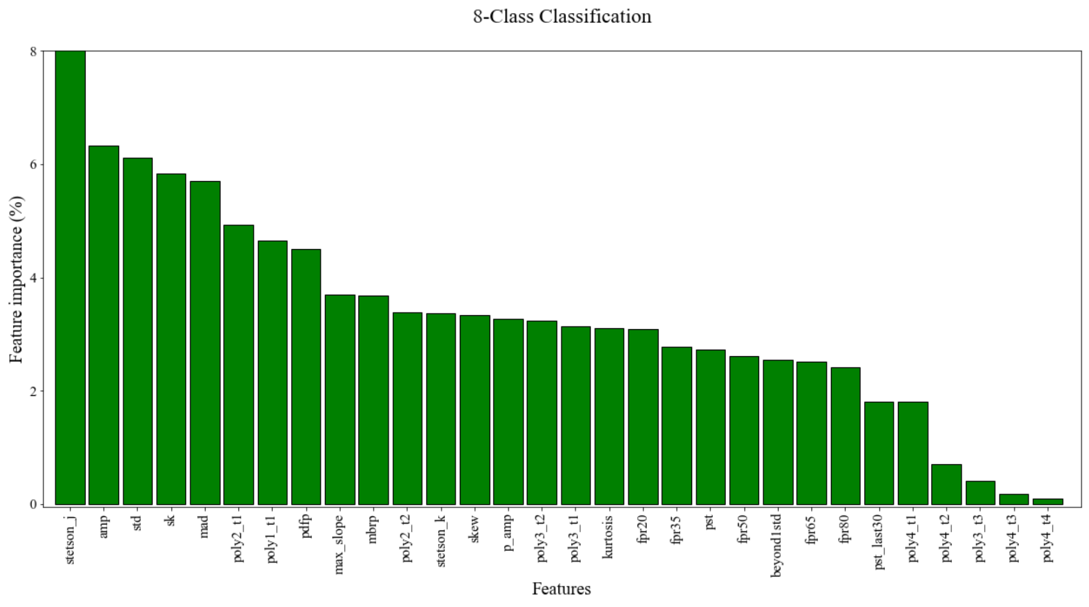

Figure 2. Feature importance rank for the best Random Forest classifier for the best 8-Class classification task. Feature importance is represented with percentages.

development and exploration of automatized ways to detect transient sources. In this paper we presented an approach for the automatic recognition of transient events with machine learning techniques.

The method we present is based on the study of light curves. We extracted its characteristic features to use them as inputs to train three different machine learning algorithms: Random Forests, Neural Networks and Support Vector Machines. The features extracted from light curves were

	AGN	Blazar	CV	Flare	HPM	Non-Transient	Other	SN
AGN	66.14	1.57	0.00	0.79	0.00	6.30	20.47	4.72
Blazar	14.29	38.57	10.00	0.00	0.00	0.00	10.00	27.14
CV	0.85	2.13	74.89	0.43	0.43	2.55	3.83	14.89
Flare	0.00	0.00	5.36	39.29	0.00	48.21	0.00	7.14
HPM	1.14	0.00	1.14	0.00	86.36	9.09	0.00	2.27
Non-Transient	0.63	0.00	1.27	1.90	1.90	84.13	5.71	4.44
Other	13.92	3.61	9.79	0.00	0.52	11.34	36.60	24.23
SN	0.63	0.63	5.40	0.95	0.00	17.46	6.35	68.57

Table 7. Confusion Matrix for the best performing model in the 8-Class task.

either statistical descriptors of the observations, or polynomial curve fitting coefficients applied to the light curves.

The machine learning algorithms performed two classification tasks. and multi-class classification of various transient classes including non-transients too. Overall, the best classifier for all tasks was the Random Forest. We studied the feature importance for this model. The most important feature was always stetson_j. The proposed coefficients corresponding to the linear terms of the quadratic and monomial curve fitting are also useful in the classification task.

We provide the code and the datasets that were used in this project. The repository containing all this data may be found in the website `https://github.com/diegoalejogm/crts-transient-recognitionSection`.

In a continuation of this project we will present in a second paper a reference dataset for astronomical transient event recognition based on images of the CATALINA survey. We will preset tests using state-of-the art deep learning techniques for transient classification. lightcurves and tests on classical machine learning algorithms

ACKNOWLEDGEMENTS

We thank Andrew Drake for sharing with us the CRTS Transient dataset used in this project. We thank Juan Pablo Reyes, Dominique Fouchez and Catalina Gómez 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

Alard C., Lupton R. H., 1998, ApJ, 503, 325
Bloom J. S., Richards J. W., 2012, Data Mining and Machine Learning in Time-Domain Discovery and Classification. pp 89–112
Cabrera-Vives G., Reyes I., Förster F., Estévez P. A., Maureira J.-C., 2017, ApJ, 836, 97
Charnock T., Moss A., 2017, ApJ, 837, L28
D'Isanto A., Cavuoti S., Brescia M., Donalek C., Longo G., Riccio G., Djorgovski S. G., 2016, MNRAS, 457, 3119
Downes R. A., Webbink R. F., Shara M. M., Ritter H., Kolb U., Duerbeck H. W., 2005, Journal of Astronomical Data, 11
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
Faraway J., Mahabal A., Sun J., Wang X., Yi Zhang L., 2016, Statistical Analysis and Data Mining: The ASA Data Science Journal, Vol.~9, Issue 1, p.~1-11, 9, 1
Gieseke F., et al., 2017, MNRAS, 472, 3101
Graham M. J., Djorgovski S. G., Mahabal A., Donalek C., Drake A., Longo G., 2012, Distributed and Parallel Databases, 30, 371
Hastie T., Tibshirani R., Friedman J., 2016, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics). Springer
Hinners T. A., Tat K., Thorp R., 2018, AJ, 156, 7
Ivezić Ž., et al., 2008, preprint, (`arXiv:0805.2366`)
Jian H.-Y., et al., 2017, ApJ, 845, 74
Jurić M., et al., 2015, preprint, (`arXiv:1512.07914`)
Kessler R., et al., 2015, AJ, 150, 172
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
Masci F. J., et al., 2017, PASP, 129, 014002
Pedregosa F., et al., 2011, Journal of machine learning research, 12, 2825
Richards J. W., et al., 2011, ApJ, 733, 10
Sako M., et al., 2008, AJ, 135, 348
Smith A. M., et al., 2011, MNRAS, 412, 1309
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