Refining Exoplanet Detection Using Supervised Learning and Feature Engineering

Margarita Bugueño
Departamento de Informática
Univ. Técnica Federico Santa María
Santiago, Chile
margarita.bugueno.13@sansano.usm.cl

Francisco Mena
Departamento de Informática
Univ. Técnica Federico Santa María
Santiago, Chile
francisco.mena.13@sansano.usm.cl

Mauricio Araya
Departamento de Informática
Univ. Técnica Federico Santa María
Valparaíso, Chile
maray@inf.utfsm.cl

***Abstract*—The field of astronomical data analysis has experienced an important paradigm shift in the recent years. The automation of certain analysis procedures is no longer a desirable feature for reducing the human effort, but a must have asset for coping with the extremely large datasets that new instrumentation technologies are producing. In particular, the detection of transit planets — bodies that move across the face of another body — is an ideal setup for intelligent automation. Knowing if the variation within a light curve is evidence of a planet, requires applying advanced pattern recognition methods to a very large number of candidate stars. Here we present a supervised learning approach to refine the results produced by a case-by-case analysis of light-curves, harnessing the generalization power of machine learning techniques to predict the currently unclassified light-curves. The method uses feature engineering to find a suitable representation for classification, and different performance criteria to evaluate them and decide. Our results show that this automatic technique can help to speed up the very time-consuming manual process that is currently done by scientific experts.**

***Keywords*—Machine Learning, Exoplanet Detection, Feature Engineering.**

I. Introduction

Planets orbiting stars outside our solar systems are called extra-solar planets or exoplanets. Detecting these planets is a challenging problem, because they only emit or reflect very dim magnitudes compared to their host stars, and they are very near to them compared to the observation distance. Several approaches have been proposed by astronomers for detecting them, being the fine-grained analysis of periodicities in star light-curves the most successful so far. However, the large volume of data that is being generated by modern observatories, including large surveys of astronomical objects, requires the use of automatized methods that can reproduce the analysis performed by astronomers to decide if the data supports the existence of an exoplanet. Fortunately, the advances in numerical methods, machine learning and data science in general allow us to apply algorithms and computational techniques that learn and predict from complex patterns in a reasonable frame of time. This paper presents how to use machine learning techniques to refine the detection of exoplanets using real data from the Kepler Space Telescope. Concretely, we explored different feature extraction and selection techniques applied to the light-curves to improve the training and prediction performance of supervised classification methods. To compare them, we used a few standard statistical learning criteria, yet the results motivated an hybrid selection of techniques that reduces the number of unconfirmed light curves.

The results of this work show that using only standard statistics to summarize the entire light curve has only a moderate performance compared to combining them with other manually extracted features by experts on astronomy related to the planet and its hosting star. We reached an 88.3% of average precision and recall (*f1 score*), meaning that approximately the 88% of the times the prediction was correct. For this configuration the best learned model was Random Forest. We combined the identification of *Confirmed* objects (exoplanets) using Random Forest and the identification of *False Positive* objects (not exoplanet) using a SVM RBF model to produce our classification.

Thus, the present work gives the classification of the Kepler Objects of Interest that have been studied by NexScI scientific staff to September of the last year 2017 (*Candidates*).

The paper is organized as follows. Section 2 presents a brief introduction to the methods of exoplanet detection which inspired this work. Section 3 discusses the datasets of Kepler Objects of Interest and the manual classification performed over this data. Section 4 introduces some machine learning methods and its corresponding metrics to use. Section 5 proposes an experimental study with empirical results on exoplanets classification and, in Section 6, we present our conclusions.

II. Background

The study of exoplanets is a relatively new field of astronomy which started with the first confirmed detection of a very fast-orbiting giant planet [1], named 51 Pegasi b. Since then, the advances in instrumentation and data analysis techniques have allowed the discovery of thousands of exoplanets. For example, NASA reports that more than 3500 exoplanet has been detected[1] using different techniques. Sadly, in mostly all the cases the currently observatories, grounder or spatial, have to applied indirect methods due to the difficult detection.

[1] http://exoplanets.nasa.gov

TABLE I
NUMBER OF CONFIRMED EXOPLANETS ACCORDING TO THE DETECTION METHOD.[3]

Astrometry	1
Direct visual detection	44
Radial velocity	676
Transit	2951
Gravitational micro lenses	64
Radio pulses of a pulsar	6

A. Why is difficult to detect exoplanets?

A planet is an object that orbits around a star and is massive enough to clear of dust and other debris the protoplanetary disk from which it was born. The theory of extrasolar planets is under development since mid-nineteenth century and although there were some unsubstantiated claims regarding their discovery, it was not until recently that we have confirmed detections. Now, we can start to answer question such as how common they are and how similar they are to the Solar system planets.

Protoplanetary disks are regions of gas and dust orbiting around young stars. Current theories suggest that the dust particles begin to collapse by gravity forming larger grains. If these discs survive to stellar radiation and comets or meteorites, the matter continues compacting giving way to a planetoid. Unfortunately, most of the discovered planets are large compared to the dimensions of the Earth, due to limitations in detection methods based on the precision of current observatories.

Planets are very dim sources of reflected light compared to their host stars. Therefore, it is extremely difficult to detect this type of light. To date, only a couple of dozen exoplanets have been photographed while the majority of known exoplanets have been detected through indirect methods. As indicated in the Table I, the most successful detection mechanisms are:

- *Radial velocity*, which studies the speed variations of a star product of its orbiting planets, analyzing the spectral lines of this one through the Doppler effect to measure the red-shift or blue-shift. This method has been successful, but it is only effective on giant planets near its star.
- *Transit photometry*, photometric observation of the star and detection of variations in the intensity of its light when an orbiting planet passes in front of it, blocking a fraction of the starlight. This method efficiently detect high-volume planets independently of the proximity of the planet to its star.

Fortunately, technological advances in photometry have allowed experiments like the space observatory Kepler to have sufficient sensitivity for detecting a greater range of exoplanets. To achieve this, feature extraction, classification and regression methods and models are needed.

Previous work

A typical source for these detections are RR Lyrae Stars, because their intensity varies through time depending on the planets. For example, Richards et al. [2] presents a catalog of variable stars and manually extracted light curve specialized features from simple statistics and other features based on the period and frequency analysis of a LombScargle fitted model [3]. Donalek et al. [4] also worked on classifying variable stars from the Catalina Real-Time Transient Survey (CRTS) and the Kepler mission, extracting similar features from the light curve to Richards et al. [2]. A different methodology was presented by Mahabal et al. [5] in which light curves are transformed into an image (grid) that represent the variations of magnitude through the variations of time intervals. They also used the data from CRTS and the task was to classify the variability of the star. The recent work of Hinners et al. [6] presents different machine learning techniques and models with the objective of classifying and predicting features over the same data that this paper uses. Similarly to what we propose here, they extract some statistical features from the light curve, but they were not interested on detecting if the light curve variations were indeed generated by an exoplanet or by another phenomena. They also tried a recurrent neural network from automatic feature extraction and prediction but with inconclusive results.

In this work we tackle the exoplanet detection problem, with ad-hoc feature extraction over the sequence in addition of automatic techniques using Fourier transform and component analysis. Then, combining the best learned model on each class, we complete the proposed task.

III. DATA

The Kepler Mission is a space observatory launched by NASA in 2009 with the goal of searching for planets similar to the size of Earth within our galaxy neighborhood. Kepler measures the variation of light from thousands of distant stars, in search of planetary transits[2]. Currently, NASA Exoplanet Science Institute has shown[3] that around 65% of exoplanet discoveries (2344) have been detected thanks to the Kepler Mission. Considering that most of the discovered exoplanets have been detected through the transit method, and taking advantage of the photometric improvements of Kepler, we propose to work with the *Kepler Objects of Interest* (KOI[4]) dataset. This dataset, provided by MAST (*Mikulski Archive for Space Telescopes*), is composed by 9564 records with 44 features each, including metadata and links to the actual light curves [7].

We collected 8054 FITS files from the archive, where some of them contain more then one light curve because different *KOI*s were detected for the same star. Moreover, each file contains the error associated to each measure, the time (in Julian date) when the measure was made, the raw light curve, the filtered light curve and a Mandel-Agol fit of the light curve [8].

[2] https://exoplanetarchive.ipac.caltech.edu/docs/KeplerMission.html
[3] https://exoplanetarchive.ipac.caltech.edu/docs/counts_detail.html
[4] http://archive.stsci.edu/search_fields.php?mission=kepler_koi

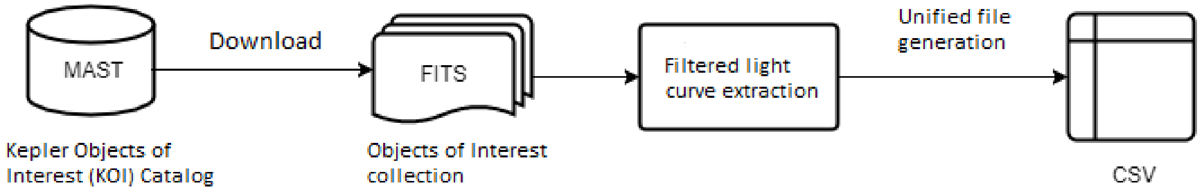

MAST
Download
FITS
Filtered light curve extraction
Unified file generation
Kepler Objects of Interest (KOI) Catalog
Objects of Interest collection
CSV

Fig. 1. Process to obtain light curves and create the data to experiment.

Every record is associated to a Kepler Object of Interest labeled as *Confirmed*, *False Positive* or *Candidate*, according to Nasa Exoplanet Science Institute[5].

- 2281 CONFIRMED: those that through extensive analysis have been confirmed as exoplanet.
- 3976 FALSE POSITIVE: those that were initially selected as candidate exoplanets but there is additional evidence that shows they are not.
- 1797 CANDIDATE: those that are still under study.

Between the reasons to catalog a candidate as a *False Positive*, according to MAST, are observation that did not match with the star position on study, showing that the transit was on another object in the background. Another possibility is that the deep of the even transit was statistically different to the deep of the odd transits, showing a binary system, i.e two stars orbiting among them.

Even though each stored light curve (Figure 2) was about 70000 measurements, every point in the series is not generated independently [9]. As the dispersion varies through time, the series is governed by a trend and could have cycles. This fact is important because the Kepler measurements are not recorded uniformly, getting light curves with missing data. On average, the missing data is about 22.98% of the size of the fully sampled light curve, as it shown in the Appendix. This means that every light curve has approximately about 55000 effective measurements. We used two simple techniques to tackle missing values: (1) fill with zeros and (2) fill the gaps with linear interpolation.

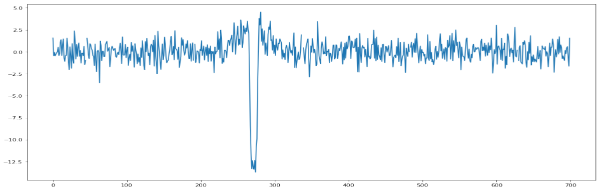


Fig. 2. Sample of a light curve as a raw format, with whitened filter applied.

From all the metadata that was available, we have selected only 10 features that we found relevant to train the models.

- *Period:* the average of the interval time between transits, based on a linear fit among all the observed transits.
- *Transit Depth:* the fraction of the stellar intensity lost on the minimum planetary transit.
- *Planet Radius:* the inferred radius of the Object of Interest.
- *Planet Teq:* the expected temperature of equilibrium on the surface of the candidate planet.
- *Stellar Teff:* is the effective stellar temperature (photosphere) in Kelvins.
- *Stellar log(g):* the logarithm of the stellar surface gravity.
- *Stellar Metallicity:* is the logarithm of the relationship between Fe and H on the surface of the star, normalized by the solar relationship between Fe and H.
- *Stellar Radius:* the stellar radius with respect to the sun (Solar = 1).
- *Stellar Mass:* the computed mass of the star.
- *KOI count:* the numbers of identified candidates on that system, which varies between 1 and 7

Following the standard dataset separation for machine learning, we used the 64% of the labeled data for training, 18% for validation and another 18% for evaluating and compare the models trained as test set. This last one represents the actual target unlabeled data that is unknown (Candidates).

Whitened filtering

The data used in this paper was the light curve on its raw format with a whitened filtered applied, this is because the objective of this filter is to obtain a light curve with a constant white noise where the higher signal (high to noise) get amplified and give a more uniform signal. Whitened filtering is a linear transformation that take a sequence of random variable (with know covariance matrix) into a new sequence of new random variables where the covariance matrix is the identity, no correlation between variables and variance normalized. The transformation is called whitened because change the input data into a new data with white noise. The white noise is a random signal that have the same intensity on different frequencies, which gives a spectral density of constant power. The operation realized over the light curve it is divided the signal by his own spectral power density function.

Mandel-Agol model

Within the FITS files there is also the Mandel-Agol model fitted to the light curve, which model the transit of a stratospheric planet around a stratospheric star, like an eclipse, assuming a uniform source. It requires to know the distance from the center of the planet to the center of the parent star as well as the radius of each one of the bodies. Mandel-Agol

[5] http://nexsci.caltech.edu/

models the opacity observed on the light intensity according to the planet position. When the planet eclipse the star the opacity is maximum, when the planet orbits without eclipse the star the opacity is minimum and uniform (zero or null), yet, when the planet is close to eclipse the star the intensity is modeled as a quadratic polynomial according to [8].

IV. Models and Methods

The used models aim to catalog correctly the objects that are currently investigated by NexScI (*Candidate*) labeling them as *Confirmed* or *False Positive* according to what the model learns.

A. Feature extraction

In this work we use two different methods for feature extraction. The first one focuses on the use of manual techniques for feature extraction and construction, and the second one focuses on automatic techniques for the same propose: feature generation. Both methods are applied to the processed light curve mentioned in the above section.

1) Manual feature extraction: We used extraction techniques specialized on time series, which in this case corresponds to measurements of intensity of the light along time, inspired on the library *Feature Analysis for Time Series* (*FATS*) for Python [10]. This library was created with the purpose of extracting features over astronomical data (light curves specially) and was used before over the same data [6]. Besides this, features have been used on other tasks over light curves [2], [4]. Due to performance issues of the library[6], we have developed our own implementation for some of the features present in the package.

- *Amplitude*, defined as the difference between the maximum value and the minimum value divided by 2.
- *Slope*, defined as the the slope of a linear fit to the light curve.
- *Max*, the maximum value of the sequence.
- *Mean*, the average of the sequence.
- *Median*, the median magnitude of the sequence.
- *Median Abs Dev*, defined as the median of the difference between every point to the median of the sequence.
- *Min*, the minimum value of the sequence.
- *Q1*, the first quartile of the data in the sequence.
- *Q2*, the second quartile of the data in the sequence.
- *Q31*, the difference between the third and first quartiles.
- *Residual bright faint ratio*, is the rate between the residual of the fainter intensities over the brighter intensities, with the mean of the sequence as threshold.
- *Skew*, defined as a measure of the asymmetric of the sequence (third standardized moment).
- *Kurtosis*, defined as the fourth standardized moment of the sequence.
- *Std*, standard deviation of the sequence.

Due the few features generated over the sequence that can summarize the light-curve (from a total of 55 thousand effective measurements of intensity of light approximately), we decided to consider some of the metadata presented in the above sections, which can can contribute with additional information of every one of the observation of the objects of interest and their light curves.

2) Automatic feature extraction: Alternatively for the automatic feature extraction techniques we use unsupervised learning methods, where the objective is to find intrinsic patterns among all the data independently from task, in this case the exoplanet detection. The first method is the well-know Principal Component Analysis (**PCA**) [11], [12]. This algorithm is a linear method that projects the data into a lower dimensional space, i.e. transforms the data space from the original dimensions (the length of the sequence) into a new space of lower dimensionality defined by the vector of higher variances. PCA is know as one of the best algorithm for dimensionality reduction and has been applied to several applications, obtaining particularly good results on time series [13]–[15]. Besides its great efficiency when dealing with high dimensional data, PCA can benefit from specific optimizations over linear algebra methods that are present on several libraries.

A second method is **FastICA** (*Fast algorithm for Independent Component Analysis*) [16], [17], an efficient iterative algorithm that finds statistically independent components of the data, in contrast to the uncorrelated ones used by PCA. The algorithm is focused on the signal abstraction, since it tries to detect the independently sources that, mixed, produce the observed data.

These two automatic methods were tested using also the Discrete Fourier transform [18], [19] applied to the light curve: it transforms the data from the time domain in which the measurements where obtained, to the frequency domain where the signal was generated. This method is designed to analyze periodic signals, which is exactly the case of transit light curves.

B. Learning Models

The K-nearest neighbors model (k-**NN**) is a popular yet simple approach based on memory (i.e., non-parametric). It remembers all the training data and uses a k number of nearest samples of the data to predict a class [20]. This algorithm classifies based on the voted majority among his k nearest neighbors based on a distance metric, in our case Minkowski's [7]. Despite its simplicity, it shows a very good performance in several problems [21].

The second model is the regularized logistic regression: a variant of the logistic regression proposed in [22] that classify based on a probabilistic (logistic) binary model in which a linear boundary is defined among the classes based on the probability of belonging to each class. The regularization is used to penalize the parameters/weights to avoid overfitting, i.e., learning more complex patterns than the ones present original phenomenon for the sake of reducing the error.

[6] https://github.com/isadoranun/FATS

[7] http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric

Another linear algorithm, namely the Support Vector Machine (SVM) [23] is a margin-based model which also defines a linear boundary among the classes and tries to find the best separation hyperplane that divide them. SVM uses a subset of the data to fit the model: only those that are closely enough to the boundary are remembered and they are called support vectors. We use the regularized version of SVM, c-SVM, that penalizes the error on the training data, similarly to what was used for logistic regression, producing better generalization results. For both the l_2 norm was selected (ridge regression). For the c-SVM we use a Gaussian kernel as the *Radial Basis Function* (RBF) [15] because it produce a more flexible decision boundary. Indeed, it computes the operations in a higher dimensional space, translating the problem to a non linear decision space. This technique brings improvements when the data is not linear separable.

As last, we use a Random Forest classifier [24], [25], an ensemble in which many models (in this case decision trees) are trained over different samples of the data (bootstrap). Every model is trained by selecting some features randomly and dividing the samples according to this features. The predictions of the ensemble corresponds to the majority class among all the models.

As an alternative to these linear models and feature extraction techniques, we use state-of-the-art recurrent neural network models, specifically the LSTM (Long Short Term Memory) [26] and GRU (Gated Recurrent Unit) [27], which are specifically designed for time series. The models considers that a sequence has a local dependency that affects the output, so it take the dependency into account. These models are designed to cope with very large sequences, because these *gates* can detect pattern while forgetting samples that are useless and keep those that are not. Given the difficult of training this network with the length of the sequence that we used, the light curve was transformed into a sequence of statistics by windows. We used: maximum, minimum, mean, standard deviation and the third moment.

C. Metrics

The exoplanet problem is an instance of unbalanced binary classification. Therefore we need to select quality measures beyond the classical accuracy, namely *precision*, *recall* (by class) and *f1 score*, where the last one summaries *precision* and *recall* metrics in just one over all the data (both classes).

- **Precision**
  Rate between the objects correctly labeled as one class over the sum of all the objects labeled as that class. In other words, is the ability of the model to label one class A only when the object effectively was from that class.
  $$P = \frac{T_p}{T_p + F_p} \tag{1}$$
- **Recall**
  Rate between the objects correctly labeled as one class over the sum of all the objects effectively from that class. In other words, is the ability of the model to include all the object that effectively are from one class A.
  $$R = \frac{T_p}{T_p + F_n} \tag{2}$$
- **F1-score**
  This is defined as the harmonic mean between the two measures previously mentioned, being high when both are high. Note that the relative contribution of precision and recall to the F1 score are equal. Therefore, this is a good quality measure of a model:
  $$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{3}$$

With T_p as the *true positive*, F_p as the *false positive* and F_n as the *false negative*. All this metrics reach their best values at 1 and worst at 0.

V. Experiments and results

Due to the large amount of data that was processed, it was necessary to use a cluster provided by ChiVO[8] (Chilean Virtual Observatory) in which 6257 labeled data, corresponding to 121 GB, and 1797 not labeled data (candidates), corresponding to another 33 GB, were downloaded. Thus, approximately 4000, 1000 and 1000 registers were grouped as training, validation and testing sets respectively. The validation set was used to tune structural hyper-parameters of the different algorithms while testing set was used to compare the best models to simulate how these will behave on future data.

Regarding the hyper-parameters of the trained algorithms, it was necessary to define:

- k-NN: Number of neighbors k.
- Logistic Regression: C, Inverse of regularization strength. Smaller values specify stronger regularization.
- SVM: C, Inverse of regularization strength. Smaller values specify stronger regularization.
- Random Forest: The maximum depth of all the trees.

Note that the selection of such hyper-parameters was not expensive in computational terms because it was performed on the representations of features extracted from the techniques already discussed, which are much smaller than the amount of data ($d \ll n$, where n indicates the data size and d the dimensionality of these).
For automatic feature extraction techniques, fixed dimensions were experimented (**5**, 10, 15, 20, 25, 50 for ICA and **5**, 10, 25, 55, 100 for PCA), where Table II shows how the performance of the best model varies according to the dimensionality. It is evident that as the dimensionality increases (characteristics), the error increases.

The use of a discrete Fourier transform seems to be a crucial procedure when extracting features automatically. If we apply the learning models to the raw data representation (sequence of intensity of light), the error turned out equally to a random

[8] http://www.chivo.cl

TABLE II
F1 SCORE OF THE BEST CLASSIFIER, RANDOM FOREST, IN FUNCTION OF DIMENSIONALITY.

	5	**10**	**15**	**20**	**25**	**50**
ICA	**0.711**	0.709	0.709	0.686	0.679	0.675

	5	**10**	**25**	**55**	**100**	**255**
PCA	**0.713**	0.701	0.701	0.699	0.702	0.689

labeling (i.e., all the examples as *false positive* class 0.486, while 0.200 for *confirmed* class in terms of *f1 score*).

Surprisingly enough, completing missing data with zeros produces a consistent improvement of $\sim$ 0.1 in the f1-score, while linear interpolation produced worse results. We tried another missing-value treatment technique consisting in performing a *sampling* of the sequence by taking the maximum value each 3 points (considering the missing data as zeros) completing the data following the trend line. Unfortunately, this resulted in a greater error.

In addition to extracting features of the light curve manually, we worked directly with the MAST metadata, results that can are shown in Table III. First, the metadata corresponding to the potential exoplanet under study (KOI - *kepler object of interest*) was used. This contains the orbit period, the transit depth, the planet radius, the equilibrium temperature of the planet and the number of KOI under study in such system. This approach proved to be better than the techniques that faced the raw data. Also, we included the metadata of the hosting star such as the effective temperature, the metallicity, the gravity, the radius and its mass. When we used these features, it was possible to notice an important improvement, because we fed the classification algorithms with more relevant information than using only features extracted from the light curve. As an alternative result, these manual processes were mixed, i.e metadata was used in conjunction with the manually extracted features from the light curve.

With the purpose of handling the unbalance data, we experimented with the undersampling technique [28], in which the majority class was subsampled, getting two sampled sets of similar sizes as a training set. Nevertheless, we also used the unbalanced data directly when we trained the Logistic Regression, SVM and Random Forest models, because they admit weighting different classes into the objective functions in a way that the minority have more impact on the objective function [29]. Over this last experiment we improved the results by $\sim$ 0.1 on f1 score metric.

The results are shown by Table III, which presents *F1 score* metric for the best representations of each technique, with 5 components for PCA and ICA, filling with zeros the missing data and assigning balanced weights to the classes (without subsampling the major class).

After testing all the variants in the experimentation process (Table III), the best result was obtained using manual techniques, in which statistics and fixed light-curve

TABLE III
F1 SCORE ON THE CLASSIFICATION OF DIFFERENT MODELS (LEARNERS) OVER THE TEST SET ON THE DIFFERENT REPRESENTATIONS GENERATED.

	Learners			
	k-NN	*Logistic Regression*	*SVM RBF*	*Random Forest*
Fourier + PCA	0.679	0.493	0.486	**0.713**
Fourier + ICA	0.679	0.493	0.486	**0.711**
OwnFATS	**0.666**	0.583	0.575	0.658
Planet metadata	0.825	0.848	0.848	**0.870**
Stellar metadata	0.766	0.718	0.751	**0.766**
OwnFATS + stellar & planet metadata	0.844	0.864	0.876	**0.883**

features were extracted in addition to other features based on metadata (both for the planet and the star). We obtained a performance of 88.3% for future classification, as *F1 score* metric indicated, meaning that probably 88.3% of the time the predictions will be correct. The best trained model was Random Forest, where the Figure 3 presents the feature importance/relevance that this ensemble has on his process of selection in the prediction. In the figure, it can be seen that the most relevant features belong to the object in study (object of interest). Particularly, the radius of the object is the one generate most impact on the prediction together with the period and the number of object in studies in that system. The features with less importance happens to be the ones extracted from the light curve, where the slope and second quartile are the ones with less impact.

The experimentation with recurrent neural network was very complete but without success: we tested with different representation on the input data, we varied the size of the window from 300 to 500, we varied the architecture of the network modifying the depth, changed the number of units, tried different optimizer (RMSprop and Adam), used different number of epochs as well as modified the batch size during the training. Unfortunately, no good results were achieved for any the networks, being a little network with GRU the one with the best performance, 0.567 according to *f1 score*. Knowing the good performances of these models in other areas, we suspect that the sequence was too long and by that, the statistic by window was not the best technique to summarize all the needed information for the proper learning of the network.

The details of the precision and recall metrics on the classification over both classes on the test set can be found in the appendix. In Table VI we report the classification on the *false positive* class. It can be seen that the best model that can identify correctly this class based on this two metrics, i.e. the model that cover the most examples of that class and do this in a meticulous way (without including examples from other class), is the SVM RBF. This can be explained due to the RBF kernel, because it can fit boundaries on a flexibly way and very tight to the data of that class. Similarly, in this

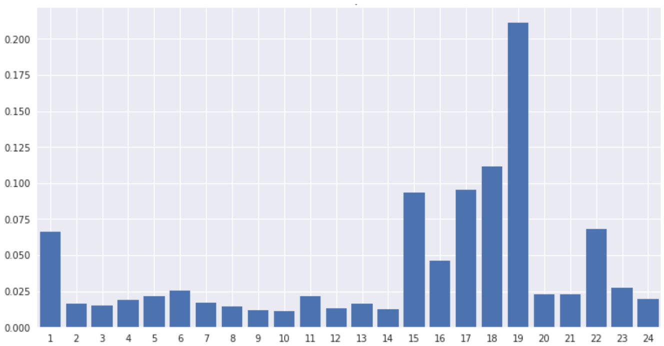


Fig. 3. Feature importance of Random Forest model over the representation with the best performance (OwnFATS with metadata). The names of the indexes features are in Table IV.

TABLE IV
INDEXES FEATURES OF FIG. 3. THE MOST IMPORTANT ARE IN BOLD.

1	minimum	9	Q2	17	**teq**
2	maximum	10	slope	18	**koi count**
3	mean	11	amplitude	19	**planet radius**
4	std	12	median absolute deviation	20	teff
5	IQR	13	residual bright-faint ratio	21	log(g)
6	skewness	14	median	22	metallicity
7	kurtosis	15	**period**	23	stellar radius
8	Q1	16	transit depth	24	stellar mass

table it can be seen that the ICA transformation applied over the frequency domain achieve the higher precision among all the techniques of features extraction, even the metadata, but with the trade-off of a small recall. This mean that it can only cover a small portion of the data of that class. In the same way, we can analyze the Table VII, i.e., the classification on the *Confirmed* class, showing lower scores that the other class, suggesting the difficulty on the exoplanet prediction problem. This could be because all of them do not have very similar features on the light curve, making it difficult to detect and group all the samples of this class. However, the best model in the task of only detecting exoplanets (*Confirmed*) on the different representation of the data was Random Forest.

Final results: After having performed all the corresponding test and identified the best model on each label over the future data based on the metrics precision and recall, we show the predictions on the representation **OwnFATS + stellar & planet metadata** with the Random Forest model for those *Confirmed*, with maximum depth 15 and the SVM RBF model with regularization parameter 100 for those *False Positive*. Therefore, we show the classification over the Kepler Object of Interest that are still being studied by the staff of NexScI on September of 2017, i.e. those object labeled as *Candidate*. On the Table VIII (in the appendix) we show a sample of the labels predicted by the models mentioned, being these *Confirmed* or *False Positive* as appropriate. Also, we report when the models disagree on assigning labels for the same object, labeling them as *Unclassified*, because the models cannot reach a consensus. This table shows, for example that the system of the star Kepler 279, sheltering two confirmed exoplanet by NexScI (Kepler 279 b and Kepler 279 c), our models show that the third object in study **K01236.04** happens to be a valid exoplanet as the others orbiting the parent star. An opposite case is the system of the star Kepler 619, which also shelters two exoplanets (Kepler 619 b y Kepler 619 c): our models assign that the third object in study **K00601.02** as a false positive. Also our models classify an entire system on study (**K01358.01**, **K01358.02**, **K01358.03** and **K01358.04**), tagging every object there as a valid exoplanet. This also can be seen when it tagged objects of interest on star with confirmed exoplanets, such as Kepler 763 b orbiting Kepler 763, where it assigned two new brother exoplanets orbiting the same star yet the fourth object on study **K01082.02** is a false positive.

Finally we present a summary table on the assignment/labeling of the models in our work:

TABLE V
SUBTOTAL OF CANDIDATE EXOPLANETS BY CORRESPONDING METHOD.

Total *CANDIDATE*	**Learner**	**1791**
Subtotal *CONFIRMED*	Random Forest	975
Subtotal *FALSE POSITIVE*	SVM RBF	434
Unclassified		382

VI. CONCLUSIONS

We introduced a new refining method to decide whether or not an object on study (KOI) is really an exoplanet using supervised automatic learning. By using different techniques focused on handling raw data (sequence of intensity of light) on machine learning and combining them properly, we reproduce the arduous and extensive work that experts perform when detecting or disconfirming an exoplanet on study. Based on the results of the feature engineering proces we indicate that the automatic techniques used to extract information from the light curve wasn't good enough compared to the metadata, which outperforms it respect to the score. The reason we find to that is the not suitable methods for the feature extraction, or well, too simple for the complex problem that we faced, because the light curves were very diverse in their morphology. Also the problem was complex regarding the execution time, because the computational cost of compute all the operations over the very long sequence and the features of the problem. The varies about which metadata used could have great impact on the informed results of the work, because the choice was made only by looking at the description of the features informed by MAST. This election could be modified achieved even better results having the right supported by an expert in that area.

About the future work, firstly is to use the fit Mandel-Agol light curve because of his smoothness. Also is the modification of the fill in techniques on the missing values or some techniques that can handle this properly. In the same way, used new and different techniques on the task of the feature extraction on the light curve.

Acknowledgments

This work was possible thanks to Chilean Virtual Observatory, ChiVO. Also we thanks to the academic Ricardo Ñanculef of the Departamento de Informtica of the Univ. Tcnica Federico Santa Mara by his contribution to this work.

References

[1] M. Mayor and D. Queloz, "A jupiter-mass companion to a solar-type star," 1995.

[2] J. W. Richards, D. L. Starr, N. R. Butler, J. S. Bloom, J. M. Brewer, A. Crellin-Quick, J. Higgins, R. Kennedy, and M. Rischard, "On machine-learned classification of variable stars with sparse and noisy time-series data," *The Astrophysical Journal*, vol. 733, no. 1, p. 10, 2011.

[3] N. R. Lomb, "Least-squares frequency analysis of unequally spaced data," *Astrophysics and space science*, vol. 39, no. 2, pp. 447–462, 1976.

[4] C. Donalek, S. G. Djorgovski, A. A. Mahabal, M. J. Graham, A. J. Drake, T. J. Fuchs, M. J. Turmon, A. A. Kumar, N. S. Philip, M. T.-C. Yang *et al.*, "Feature selection strategies for classifying high dimensional astronomical data sets," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 35–41.

[5] A. Mahabal, K. Sheth, F. Gieseke, A. Pai, S. G. Djorgovski, A. Drake, M. Graham *et al.*, "Deep-learnt classification of light curves," *arXiv preprint arXiv:1709.06257*, 2017.

[6] T. Hinners, K. Tat, and R. Thorp, "Machine learning techniques for stellar light curve classification," *arXiv preprint arXiv:1710.06804*, 2017.

[7] W. D. Pence, L. Chiappetti, C. G. Page, R. Shaw, and E. Stobie, "Definition of the flexible image transport system (fits), version 3.0," *Astronomy & Astrophysics*, vol. 524, p. A42, 2010.

[8] K. Mandel and E. Agol, "Analytic light curves for planetary transit searches," *The Astrophysical Journal Letters*, vol. 580, no. 2, p. L171, 2002.

[9] M. Falk, F. Marohn, R. Michel, D. Hofmann, M. Macke, C. Spachmann, and S. Englert, "A first course on time series analysis: Examples with sas [version 2012. august. 01]," 2012.

[10] I. Nun, P. Protopapas, B. Sim, M. Zhu, R. Dave, N. Castro, and K. Pichara, "Fats: Feature analysis for time series," *arXiv preprint arXiv:1506.00010*, 2015.

[11] K. Pearson, "Principal components analysis," *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, vol. 6, no. 2, p. 566, 1901.

[12] L. I. Kuncheva and W. J. Faithfull, "Pca feature extraction for change detection in multidimensional unlabeled data," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 69–80, 2014.

[13] M. R. Gamit, P. Dhameliya, and N. S. Bhatt, "Classification techniques for speech recognition: A review," *vol*, vol. 5, pp. 58–63, 2015.

[14] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction." in *IWANN*, vol. 5. Springer, 2005, pp. 758–770.

[15] L. Cao, K. S. Chua, W. Chong, H. Lee, and Q. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1, pp. 321–336, 2003.

[16] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4, pp. 411–430, 2000.

[17] K. Bae, S. Noh, and J. Kim, "Iris feature extraction using independent component analysis," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 838–844.

[18] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[19] A. Grinsted, J. C. Moore, and S. Jevrejeva, "Application of the cross wavelet transform and wavelet coherence to geophysical time series," *Nonlinear processes in geophysics*, vol. 11, no. 5/6, pp. 561–566, 2004.

[20] B. Dasarathy, "Nearest neighbor norms: Nn pattern classification techniques," 1991.

[21] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *Granular Computing, 2005 IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 718–721.

[22] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242, 1958.

[23] V. Vapnik, *Statistical learning theory. 1998*. Wiley, New York, 1998.

[24] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[25] L. Fraiwan, K. Lweesy, N. Khasawneh, H. Wenz, and H. Dickhaus, "Automated sleep stage identification system based on time–frequency analysis of a single eeg channel and random forest classifier," *Computer methods and programs in biomedicine*, vol. 108, no. 1, pp. 10–19, 2012.

[26] S. Hochreiter and J. Schmidhuber, "Lstm can solve hard long time lag problems," in *Advances in neural information processing systems*, 1997, pp. 473–479.

[27] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[28] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003, pp. 1–8.

[29] G. King and L. Zeng, "Logistic regression in rare events data," *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

Appendix

TABLE VI
Scores of Precision (P) and Recall (R) on the *False Positive* class of learned models on the test set over the different representation on the data. In bold is the best model on each representation.

	Learners			
	k-NN	*Logistic Regression*	*SVM RBF*	*Random Forest*
Fourier + PCA	P: 0.726 R: 0.809	P: 0.902 R: 0.283	P: 0.629 R: 1.000	**P: 0.789** **R: 0.736**
Fourier + ICA	P: 0.752 R: 0.728	P: 0.899 R: 0.285	**P: 0.933** **R: 0.332**	P: 0.788 R: 0.730
OwnFATS	P: 0.743 R: 0.695	P: 0.821 R: 0.441	**P: 0.890** **R: 0.395**	P: 0.827 R: 0.569
Planet metadata	P: 0.863 R: 0.857	P: 0.917 R: 0.830	**P: 0.927** **R: 0.817**	P: 0.914 R: 0.871
Stellar metadata	P: 0.781 R: 0.886	P: 0.806 R: 0.714	P: 0.818 R: 0.768	**P: 0.819** **R: 0.803**
OwnFATS + stellar & planet metadata	P: 0.860 R: 0.899	P: 0.919 R: 0.857	**P: 0.934** **R: 0.861**	P: 0.924 R: 0.884

TABLE VII
Scores Precision (P) and Recall (R) to the *Confirmed* class of learned models on the test set over the different representation on the data. In bold is the best model on each representation

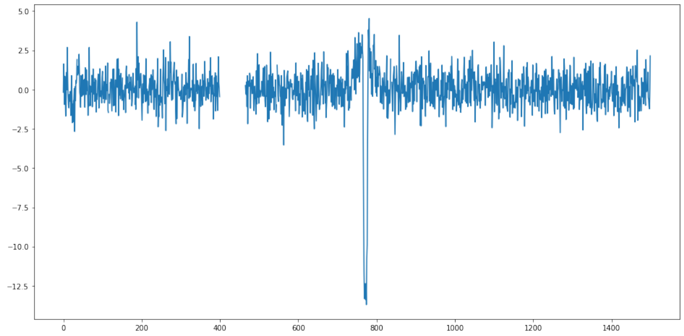
	Learners			
	k-NN	*Logistic Regression*	*SVM RBF*	*Random Forest*
Fourier + PCA	P: 0.597 R: 0.481	P: 0.438 R: 0.948	P: 0.000 R: 0.000	**P: 0.597** **R: 0.666**
Fourier + ICA	P: 0.562 R: 0.592	P: 0.438 R: 0.945	P: 0.458 R: 0.960	**P: 0.592** **R: 0.665**
OwnFATS	**P: 0.533** **R: 0.592**	P: 0.468 R: 0.836	P: 0.471 R: 0.917	P: 0.522 R: 0.798
Planet metadata	P: 0.763 R: 0.773	P: 0.755 R: 0.874	P: 0.745 R: 0.893	**P: 0.801** **R: 0.864**
Stellar metadata	**P: 0.755** **R: 0.585**	P: 0.600 R: 0.713	P: 0.649 R: 0.716	P: 0.681 R: 0.703
OwnFATS + stellar & planet metadata	P: 0.818 R: 0.756	P: 0.785 R: 0.874	P: 0.795 R: 0.898	**P: 0.820** **R: 0.879**

TABLE VIII
This table show some of the predictions assigned as much by **OwnFATS + stellar & planet metadata** with Random Forest classifier for the task of exoplanet detection (*Confirmed*), as by **OwnFATS + stellar & planet metadata** with SVM RBF for the task of non-exoplanet detection (*False Positive*). It should be mentioned that the column **Confirmed on that system** count the ammount of confirmed exoplanets on the date of the study (September 2017), as long as **Star** is the name of the Parent star in the system; The ones with no information doesn't show this value. More can be found on https://github.com/FMena14/ExoplanetDetection

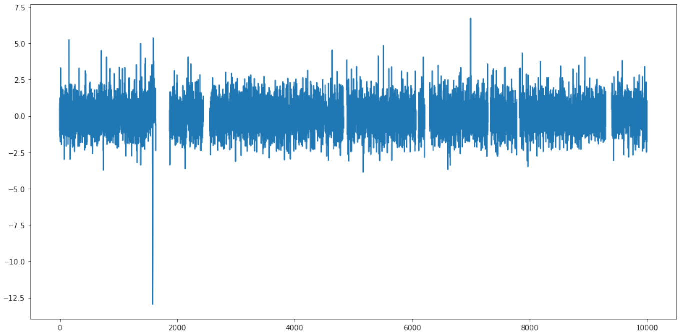
KOI name	**Disposition**	**Confirmed on that system**	**Star**
K00601.02	*FALSE POSITIVE*	2/3	Kepler 619
K00750.02	*UNCLASSIFIED*	1/3	Kepler 662
K01082.01 K01082.02 K01082.04	*CONFIRMED* *FALSE POSITIVE* *CONFIRMED*	1/4	Kepler 763
K01236.04	*CONFIRMED*	2/3	Kepler 279
K01358.01 K01358.02 K01358.03 K01358.04	*CONFIRMED* *CONFIRMED* *CONFIRMED* *CONFIRMED*	0/4	-
K01750.02	*CONFIRMED*	1/2	Kepler 948
K02064.01	*UNCLASSIFIED*	0/1	-
K02420.02	*CONFIRMED*	1/2	Kepler 1231
K02578.01	*FALSE POSITIVE*	0/1	-
K02828.02	*FALSE POSITIVE*	1/2	Kepler 1259
K03444.03	*UNCLASSIFIED*	0/4	-
K03451.01	*UNCLASSIFIED*	0/1	-
K04591.01	*FALSE POSITIVE*	0/1	-
K05353.01	*FALSE POSITIVE*E	0/1	-
K06267.01	*CONFIRMED*	0/1	-
K06983.01	*CONFIRMED*	0/1	-
K07279.01	*CONFIRMED*	0/1	-
K07378.01 K07378.02	*CONFIRMED* *CONFIRMED*	0/2	-
K07434.01	*FALSE POSITIVE*	0/1	-
K08082.01	*CONFIRMED*	0/1	-

Samples of the input data

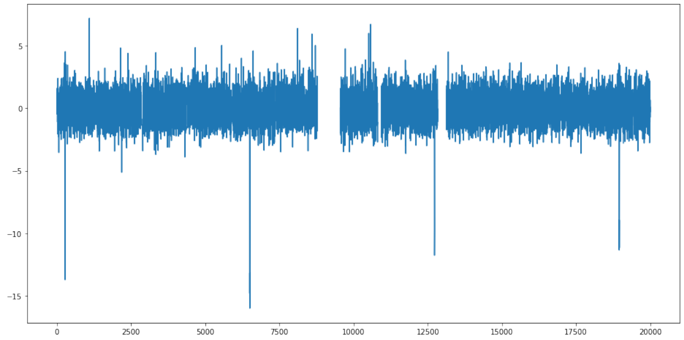
Light curve in raw data



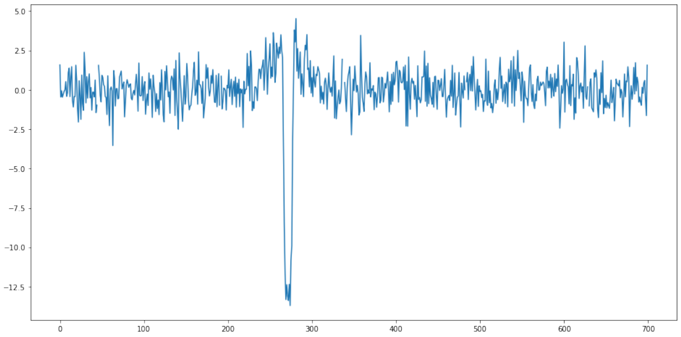
Light curve 1



Light curve 2

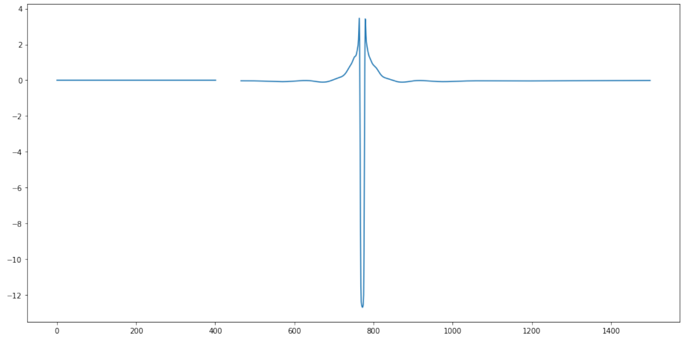


Light curve 3

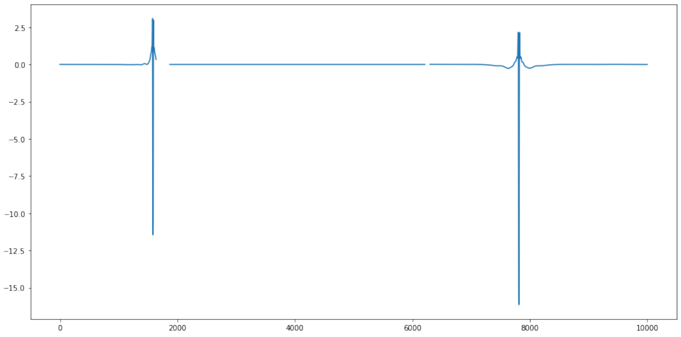


Light curve 4

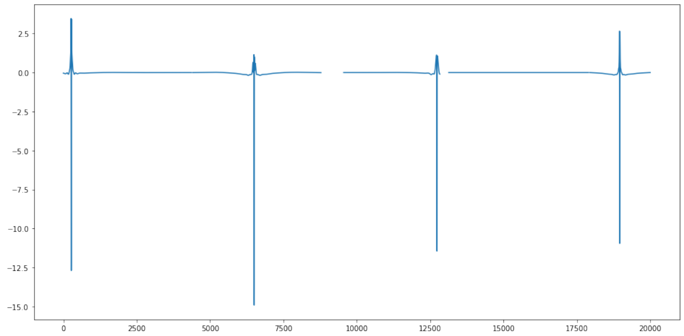
Mandel-Agol light curve fit



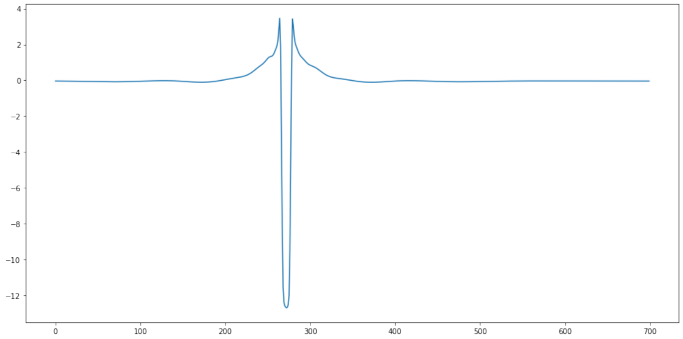
Model 1



Model 2



Model 3



Model 4