Astronomy & Astrophysics manuscript no. output
February 26, 2019

FliPer$_{\rm Class}$: In search of solar-like pulsators among TESS targets

L. Bugnet[1,2], R. A. García[1,2], S. Mathur[3,4], G. R. Davies[5,6], O. J. Hall[5,6], M. N. Lund[6], and B. M. Rendle[5,6]

[1] IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France
e-mail: lisa.bugnet@cea.fr
[2] AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, F-91191 Gif-sur-Yvette, France
[3] Instituto de Astrofísica de Canarias, E-38200, La Laguna, Tenerife, Spain
[4] Universidad de La Laguna, Dpto. de Astrofísica, E-38205, La Laguna, Tenerife, Spain
[5] School of Physics and Astronomy, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
[6] Stellar Astrophysics Centre, Department of Physics and Astronomy, Aarhus University, Ny Munkegade 120, DK-8000 Aarhus C, Denmark

Received / Accepted

ABSTRACT

The NASA's Transiting Exoplanet Survey Satellite, TESS, is about to provide full-frame images of almost the entire sky. The amount of stellar data to be analysed represents hundreds of millions stars, which is several orders of magnitude above the amount of stars observed by CoRoT, *Kepler*, or K2 missions. We aim at automatically classifying the newly observed stars, with near real-time algorithms, to better guide their subsequent detailed studies. In this paper, we present a classification algorithm built to recognise solar-like pulsators among classical pulsators, which relies on the global amount of power contained in the PSD, also known as the FliPer (Flicker in spectral Power density). As each type of pulsating star has a characteristic background or pulsation pattern, the shape of the PSD at different frequencies can be used to characterise the type of pulsating star. The FliPer Classifier (FliPer$_{\rm Class}$) uses different FliPer parameters along with the effective temperature as input parameters to feed a machine learning algorithm in order to automatically classify the pulsating stars observed by TESS. Using noisy TESS simulated data from the TESS Asteroseismic Science Consortium (TASC), we manage to classify pulsators with a 98% accuracy. Among them, solar-like pulsating stars are recognised with a 99% accuracy, which is of great interest for further seismic analysis of these stars like our Sun. Similar results are obtained when training our classifier and applying it to 27 days subsets of real *Kepler* data. FliPer$_{\rm Class}$ is part of the large TASC classification pipeline developed by the TESS Data for Asteroseismology (T'DA) classification working group.

Key words. asteroseismology - methods: data analysis - stars: oscillations

1. Introduction

Starting with CoRoT, and showing its full potential with *Kepler*, asteroseismology is now the most precise way to obtain estimates of masses and radius of field stars (e.g. Lebreton & Goupil 2014), excepting eclipsing binaries for which spectroscopy prevails. Asteroseismic parameters such as the frequency of maximum power $\nu_{\rm max}$ and the large frequency separation $\Delta\nu$ of the oscillation modes of solar-like pulsators (i.e. with modes excited by turbulent convection, Goldreich & Keeley 1977) are obtained from the power density spectrum by using global seismic pipelines (e.g. Mosser & Appourchaux 2009; Huber et al. 2009; Mathur et al. 2010, etc.). These global seismic parameters are key constraints for stellar evolution models: using them leads to age estimates that are much more precise than other classical methods (e.g. Lebreton & Goupil 2014).

The Transiting Exoplanet Survey Satellite (TESS), launched on April 18th 2018, conducts a photometric survey of 90% of the sky during its two year nominal operations (Ricker et al. 2014). It will hunt extrasolar planets mostly orbiting around M type stars. The TESS fields cover 26 sky sectors, each one covering four 24° x 24° areas from the galactic pole to nearly the ecliptic plane. Each field of view remains unchanged for 27 continuous days. The satellite will specifically observe no less than two hundred thousand main-sequence dwarf stars, $30-100$ times brighter (with an apparent magnitude smaller than ~ 10, Stassun et al. 2018) than those observed by the *Kepler* satellite. All these conditions are suitable for seismic detections in Solar-like stars, mostly in high luminosity main sequence (MS) and subgiant stars (a detailed study of the potential asteroseismic yields of the TESS mission is given by Campante et al. 2016). In addition, more than 400 million stars will be observed in the full-frame images with a 30-minute observational cadence.

The first step for the large asteroseismic survey analysis is to distinguish Solar-like pulsators from all other pulsating stars. Accurate stellar classification can be computationally expensive, but efforts have been made to classify CoRoT and *Kepler* targets (Debosscher et al. 2009; Molnár et al. 2018). For example, Mathur et al. (2016b) showed three years after the end of the *Kepler* main mission that more than 800 RGs (corresponding to about 3 percent of the total amount of observed RGs) were still misclassified as cool dwarfs (see also Hon et al. *Submitted*). However,

no public real-time automatic algorithm was developed to classify stars observed during these missions. In view of the huge amount of data to be delivered by TESS, it would be advantageous to have an automatic method to classify solar-like stars, and even other types of pulsators.

FliPer is a method to estimate surface gravities (from 0.3 to 4.5 dex) or $\nu_{\rm max}$ of Solar-type stars (Bugnet et al. 2018a, 2017). It relies on the use of the global amount of power contained in the power spectrum density (PSD) of a solar-type pulsator, which depends on its evolutionary state (Mathur et al. 2011; Kallinger et al. 2016). The method is automatic, and takes advantage of a Random Forest machine learning regressor (Breiman 2001) to estimate precise surface gravities. The algorithm is trained to learn how to predict $\log g$ from thousands of precise seismic estimates made with the A2Z seismic pipeline (Mathur et al. 2010). This way, FliPer gives estimates with a precision better that when using spectroscopy alone.

Machine learning methods such as neural networks (e.g. Bai et al. 2005), decision trees based algorithms (e.g. Pérez-Ortiz et al. 2017; Veljanoski et al. 2018) or AdaBoost (e.g. Viquar et al. 2018) already give good results for characterising the stars. For instance, Hon et al. (2018) showed that they were able to distinguish core helium-burning clump stars from hydrogen shell-burning red giant (RG) stars by using a convolutional neural network. In our study, we use FliPer parameters to classify Solar-like pulsators among all pulsating stars: instead of using a regressor (see Bugnet et al. 2018a) to estimate physical parameters, we use a classifier algorithm trained with the FliPer parameters and the effective temperature of each star. After describing the data in Section 2, we explain in Section 3 how the $\text{FliPer}_{\text{Class}}$ algorithm uses FliPer parameters ($\text{F}_{\text{p},i}$) along with the effective temperature to distinguish between the different types of pulsators. Then results from the classification of TESS simulated data and of a known sample of *Kepler* main mission data are presented.

2. Data preparation

In order to test the algorithms, the T'DA working group simulated datasets of TESS observations[1] (Lund et al. 2017). We use 10,812 simulated stars that can be studied with stellar signal only (designated as "clean" data), with additional white noise ("noisy" data), or with both additional white noise and instrumental systematics ("sysnoisy" data). As systematics can be corrected (by using similar methods to the ones applied to the K2 data, Aigrain et al. 2016), we chose to focus our study on the "noisy" dataset. The description of the sample is given in Tab. 2. Part of the γ-Doradus sample is constituted of γ-Doradus/δ-Scuti hybrid stars.

To check the reliability of our methodology on real data, we also use power spectrum densities of a sample of 1,442 *Kepler* targets observed in the long cadence observation mode (corresponding to an acquisition every 30 min) and for which we know the classification. Tab. 2 displays the number of stars in the Kepler sample belonging to each classification (Reed et al. 2018; McNamara et al. 2012; Li et al. 2019; Balona et al. 2011; Balona 2013; Sachkov 2014; Smalley et al. 2015; Serenelli et al. 2017). Long-period variability stars are not represented in the *Kepler* sample, because they can be easily classified by using the effective temperature only (there is no need to use the $\text{FliPer}_{\text{Class}}$ for these stars). The *Kepler* light curves (calibrated following García et al. 2011) considered in this work were observed for approximately 4 years. This results in a much higher frequency resolution in the PSD than what is expected for most TESS targets, which are observed for only 27 days. To test our methodology on data that would be representative of the first sector of TESS data, we compute the PSD of each star based on randomly extracted 27-day periods of time from the full *Kepler* time series. We also use the effective temperatures from Mathur et al. (2017) for the sample of *Kepler* stars.

Table 1. Composition of the samples from T'DA simutalted dataset and real *Kepler* data.

Type of star	$\text{TESS}_{simulated}$	*Kepler*
Solar-like (SL)	3668	802
Subdwarf B (sdBV)	129	8
β-Cephei (β-Cep)	298	5
Slowly pulsating B-type (SPB)	1846	26
δ-Scuti	115	358
γ-Doradus (γ-Dor)	1569	202
rapidly oscillating Ap (roAp)	287	3
RRLyrae	646	36
Long-period variable (LPV)	965	0
Cepheid	1289	2

3. $\text{Fliper}_{\text{Class}}$: a tool to classify pulsating stars

FliPer (Bugnet et al. 2018a) is a method to estimate surface gravity of solar-like pulsating stars based on the measure of the amount of power in their PSD. For solar-type pulsators, the PSD is dominated by the power of the convective background, stellar oscillation modes and the signal of the rotation periods. All these effects vary when the star evolves from MS to the red giants branch (RGB). FliPer thus gives constraints on the evolutionary stage of the solar-like pulsator. We define the FliPer metric as:

$$\text{F}_{\text{p}} = \overline{\text{PSD}} - \text{P}_{\text{n}}, \tag{1}$$

where $\overline{\text{PSD}}$ represents the averaged value of the PSD from a given frequency to the Nyquist frequency, and P_{n} is the photon noise (see Bugnet et al. 2017, for more information).

3.1. FliPer parameters: $\text{F}_{\text{p},i}$

For each star we calculate different FliPer parameters, $\text{F}_{\text{p},i}$, as the FliPer metric starting from different lower frequency boundaries ($i \in [0.7, 7, 20, 50]$ μHz) in the calculation of $\overline{\text{PSD}}$. The four different frequency domains used for the $\text{F}_{\text{p},i}$ calculation are represented by the coloured area in Fig. 1.

[1] Datasets can be downloaded after registration on the TESS Asteroseismic Science Operations Center (TASOC) website at `https://tasoc.dk/wg0/SimData`

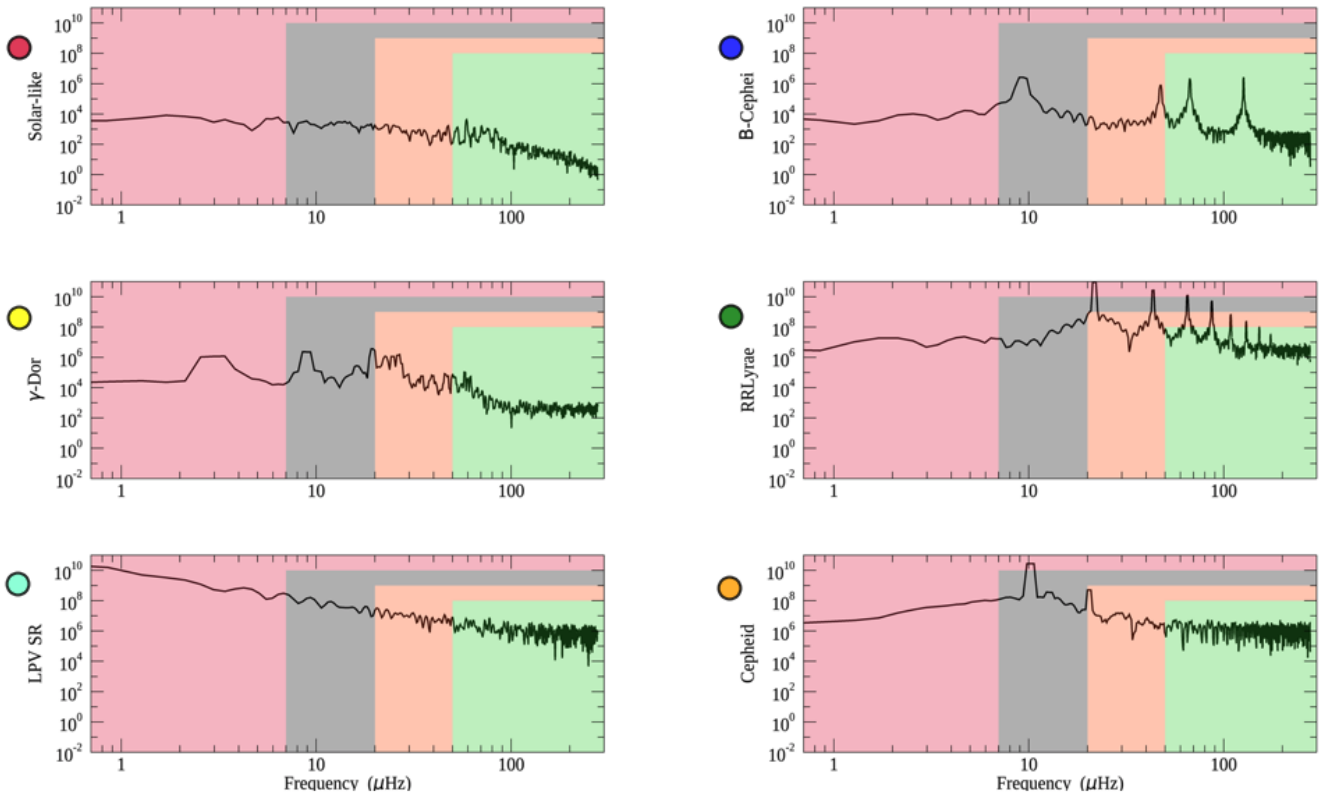

Fig. 1. PSD of six different simulated stars belonging to different classes (Solar-like, β-Cephei, γ-Dor, RRLyrae, LPV, and Cepheid) as described by the y axis labels of each panel. Coloured areas (red, grey, orange, and green) represent the different ranges of frequency used for the $F_{p,i}$ calculation (respectively from 0.7, 7, 20, and 50 μHz to the Nyquist frequency). Coloured circles represent class identifiers used in Fig.2.

By combining these different $F_{p,i}$, we extract information from different regions of the PSD of the star. A previous study (see Bugnet et al. 2017) indicates that the two $F_{p,0.7}$ and $F_{p,7}$ parameters are easily dominated by rotation peaks for MS stars, but are perfectly suitable to take into account the power of the modes for high-luminosity giants. The other parameters, $F_{p,20}$ and $F_{p,50}$, allow precise estimates for MS stars, while they do not take into account the mode power in high-luminosity RGs. FliPer gives great results when distinguishing MS stars from RGs by estimating their surface gravity, as discussed in Bugnet et al. (2018a). By combining the different $F_{p,i}$ for all stellar types, we attempt to classify not only solar-like stars but all types of pulsators.

Each type of pulsator has a typical amount of power associated for a given frequency range in the PSD. Figure 1 shows the TESS simulated PSD for 6 different types of pulsator. First, we observe that each type of star presents a characteristic signature in the PSD.

By calculating $F_{p,0.7}$ (red areas on Fig. 1), it is easy to distinguish the Solar-like star from the long period variable (LPV), as their granulation power differs by several order of magnitude. However, it is harder to distinguish the Cepheid from the RRLyrae using only $F_{p,0.7}$ as they both present a PSD background with the same order of magnitude. It is by using a higher frequency boundary such as 50 μHz for the $F_{p,i}$ calculation that the Cepheid can be well-distinguished from the RRLyrae. However, by simultaneously using the different $F_{p,i}$ it is possible to disentangle the different types of stellar pulsators.

As previously discussed, the type of variability shown by the star affects the range of values that it can have for each $F_{p,i}$. Figure 2 represents the total sample of TESS simulated data in the $\log(T_{\rm eff})$ vs. $F_{p,i}$ diagram for $i = 0.7$ μHz (Left panel) and $i = 20$ μHz (right panel) values. In addition, the stars shown in Fig. 1 are represented in the diagrams with star symbols with the same colour code as in Fig. 2. We also represent the first planet star host observed by TESS in the π-Mensae system with a star symbol (Huang et al. 2018; Gandolfi et al. 2018). This star is properly classified as a solar-like pulsator attending to its FliPer values as shown in Fig. 2.

By using the TESS simulated dataset, we notice that each type of star covers a given region of the $T_{\rm eff}$ vs. $F_{p,i}$ diagrams. This means for instance that by using only one $F_{p,i}$, solar-like pulsators are already well separated from Cepheids and from RRLyrae. However, we extend the analysis of Fig. 1 and show with Fig. 2 that using only one $F_{p,i}$ does not allow us to distinguish well between Cepheids and RRLyrae. In addition, we observe by comparing the two panels of Fig. 2 that the area corresponding to a given type of star changes when using a different $F_{p,i}$: each type of pulsator evolves differently in the diagram when modifying the starting frequency of the $F_{p,i\in[0.7,7,20,50]}$ calculation (when switching from left to right panel on Fig. 2). RRLyrae should then be separated from Cepheids

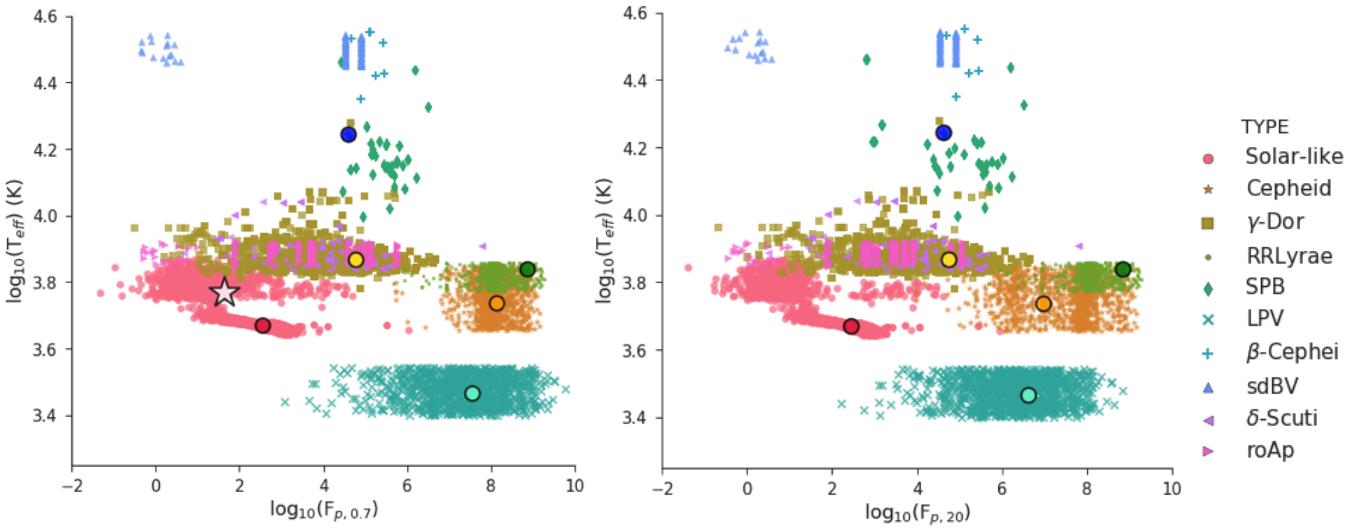

Fig. 2. Left panel: Representation of the total sample of simulated TESS stars in the $\log(T_{\rm eff})$ vs. $\log(\rm F_{p,0.7})$ diagram. Each type of stars is associated with a unique color and symbol reported on the side legend. In addition, positions of the stars represented on Fig. 1 are added to the diagram and represented with circle symbols. The white star represents the position in the diagram of the TESS target TIC 261136679. **Right panel**: Same as left panel, but for $\rm F_{p,20}$.

by comparing their different $\rm F_{p,\mathit{i}}$ ($i \in [0.7, 7, 20, 50]$ μHz).

3.2. The FliPer$_{\rm Class}$ classification algorithm

In the previous section we explained how stars can be manually classified according to their $\rm F_{p,\mathit{i}\in[0.7,7,20,50]}$. In view of the amount of TESS data to be released, the classification of each individual pulsator has to be automatic. A Random Forest classifier (Breiman 2001) is a supervised machine learning (ML) algorithm that classifies data from a given set of input parameters (See Appendix A for more details about the classifier). Random Forest algorithms were already proven to be efficient to distinguish between MS stars and RGs (Bugnet et al. 2018a) when using $\rm F_{p,\mathit{i}}$ ($i \in [0.7, 7, 20, 50]$ μHz) as input parameters.

In this study the classification of pulsators is done by using the "RandomForestClassifier" function from the "sklearn.ensemble" Python library (Pedregosa et al. 2011). We split the simulated data set into two random samples. The training set contains 80% of the total amount of stars, while the test sample contains the remaining 20% of stars. The same method is applied to the *Kepler* set. The supervised classifier FliPer$_{\rm Class}$ is trained on the training dataset to learn how to predict the output classification, by using $\rm F_{p,\mathit{i}}$ ($i \in [0.7, 7, 20, 50]$ μHz) and $T_{\rm eff}$ as input parameters for each star. The maximum number of features considered at each split point is $p = \sqrt{m}$ as we consider $m = 5$ input parameters (see Appendix A for details about the classifier). The previously trained algorithm, along with the code to use it, can be downloaded from GitHub[2]. Each parameter has a different impact on the training process, which is represented on Fig. 3 for the TESS simulated dataset. Feature importance is the number of times a feature is used to split a node normalised by the total number of nodes. Uncertainties are calculated by taking the standard deviation of each feature importance from the individual trees. The effective temperature has the largest weight to classify the type of stars. However, all input parameters are useful regarding the importances of the other $\rm F_{p,\mathit{i}}$ parameters. This shows that $\rm F_{p,\mathit{i}}$ parameters, coupled with $T_{\rm eff}$, are suited parameters to classify stars. Similar results are obtained when training on the *Kepler* training sample.

[2] https://github.com/lbugnet/FLIPER_CLASS

4. Classification of TESS simulated data

We obtain an out-of-bag (OOB) error of the training on TESS simulated data of about 0.011. This number gives estimates of the error rate of the classifier when using $n_t = 200$ trees, by classifying a sub sample of stars which were not used in the building of the last learner. The OOB error can be biased depending on the hyperparameters of the algorithm (number of trees (n_t), number of features con-

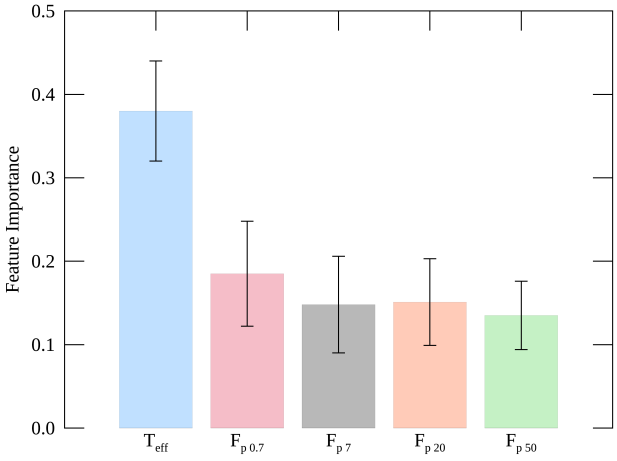

Fig. 3. Significance of the different input parameters on the training process based on the TESS simulated dataset along with their uncertainties.

Table 2. Confusion matrix of the TESS simulated data test sample. Values represent the number of stars and italic numbers in parenthesis represent the percentage accuracy for the class. The color-code is the same than on Fig. 2, and is normalized for each row by the total number of stars in each true class. Numbers that don't belongs to the diagonal represent classification errors by FliPer$_{\rm Class}$.

True \ Predicted	S-l	sdBV	β	SPB	δ	γ	roAp	RR	LPV	Ce
Solar-like	713 (*99.9*)							1		
sdBV		33 (*100*)								
β-Cep			52 (*100*)							
SPB				360 (*100*)						
δ-Scuti					18 (*64.3*)	9	1			
γ-Dor	7			3	2	320 (*96.1*)	1			
roAp	1						50 (*98.1*)			
RRLyrae								115 (*98.3*)		2
LPV									214 (*100*)	
Cepheid						1		6		256 (*96.6*)

sidered at each split point (m), etc, Mitchell 2011). This study thus provides another estimate of the classification accuracy using the TESS test sample to examine the performance of the trained algorithm.

The $\sim 2,000$ stars that belong to the TESS test sample are automatically classified amongst the classes reported in Tab. 2 by FliPer$_{\rm Class}$ trained on the training sample. Results are represented in Tab. 2: numbers in each row represent, for a given class of pulsator, the number of stars classified in each output class by FliPer$_{\rm Class}$. The higher the value on the diagonal (a high value corresponds to a dark-colored cell), the more accurate the algorithm is for the corresponding class. From this table, we first conclude that $\sim$ 98% of stars in the test set are well-characterised by the algorithm.

Then, by looking specifically at the misclassified stars, we show that most classification errors concern classical pulsators. In particular, δ-Scutis are classified as γ-Doradus, which can be explained by the fact that there are not enough δ-Scuti stars in the sample for the algorithm to learn how to recognise this type of star, and also because the γ-Doradus training sample contains some γ-Doradus/δ-Scuti hybrid stars. This misclassification problem should be solved with real TESS data as the training of the algorithm will be made on a larger set of stars belonging to each categories, including new hybrid categories, which will allow us to separate well similar classes such as δ-Scuti/γ-Dor. By looking at spectra, we notice that most misclassified solar-like and roAp stars show nearly flat power spectra, except at very low frequency. It is already known from Bugnet et al. (2018a), that FliPer, and thus FliPer$_{\rm Class}$, is not efficient for this type of noise-dominated spectrum as it compares the global amount of power in the power spectrum with the power at high frequency (representing the photon noise).

5. Classification based on 27 day segments of real *Kepler* data

To estimate the accuracy of the method on real data, we train (resp. test) the algorithm on 80% (resp. 20%) of the global set of *Kepler* data. As the number of some types of classical pulsators (such as sdBV, β-Cep, RoAp and Cepheid stars) observed by the *Kepler* main mission is very low (see Tab. 1), it is too ambitious to train and test the algorithm to recognise all different types of stellar pulsators. To avoid misclassification due to the lack of stars in the *Kepler* catalogue, we chose to group several pulsators into categories dependent upon their position in the Hertzsprung-Russell diagram:

- δ-Scuti, RoAp, and sdBV stars have low luminosity ($10L_\odot < L < 100L_\odot$).
- β-Cep and SPB stars have high luminosity ($100L_\odot < L < 100,000L_\odot$) and high effective temperatures ($4<\log_{10}(T_{\rm eff})<4.5$).
- Cepheids and RRLyrae have high luminosity ($30L_\odot < L < 100,000L_\odot$) and low effective temperature ($3.6<\log_{10}(T_{\rm eff})<3.9$).

Table 3. Confusion matrix of the *Kepler* data test sample when using $F_{p,i}$ ($i \in [0.7, 7, 20, 50]$ μHz) and $T_{\rm eff}$ as input parameters for each star. Values represent the number of stars and italic numbers in parenthesis represent the percentage accuracy for the class. The color-code is normalized for each row by the total number of stars in each true class. Numbers that don't belongs to the diagonal represent classification errors by $\text{FliPer}_{\text{Class}}$.

True \ Predicted	S-l	RR/Cep	γ-Dor	δ/roAp/sdBV	SPB/β
Solar-like	161 (*100*)				
RRLyrae/Cepheid		9 (*100*)			
γ-Dor			41 (*100*)		
δ-Scuti/RoAp/sdBV			1	73 (*97.3*)	1
SPB/β-Cephei				1	6 (*85.7*)

Table 4. Confusion matrix of the *Kepler* data test sample when using $F_{p,i}$ ($i \in [0.7, 7, 20, 50]$ μHz) only as input parameters for each star. Values represent the number of stars and italic numbers in parenthesis represent the percentage accuracy for the class. The color-code is normalized for each row by the total number of stars in each true class. Numbers that don't belongs to the diagonal represent classification errors.

True \ Predicted	S-l	RR/Cep	γ-Dor	δ/roAp/sdBV	SPB/β
Solar-like	161 (*100*)				
RRLyrae/Cepheid		8 (*88.6*)		1	
γ-Dor			41 (*99.4*)		
δ-Scuti/RoAp/sdBV	1			73 (*93.8*)	1
SPB/β-Cephei	5		1	1	0 (*0*)

We then consider the five different classes reported in Tab. 3, which represents the confusion matrix for stars in the *Kepler* test set. As in Tab. 2, values in each row represent, for a given class, the number of stars classified in each output class by the $\text{FliPer}_{\text{Class}}$. The accuracy of the classification of the *Kepler* test sample is approximately 99%. We point out that all solar-like stars are correctly classified by the algorithm (which we recall was our main goal). Most misclassifications concern classical pulsators, with a low corresponding number of stars in the training set, so the training was probably more difficult for these types of stars. This problem should be solved by training the algorithm on a much larger set of TESS observations.

5.1. Impact of the effective temperature for the classification

A very similar distribution of input parameter importances to Fig. 3 is obtained when training on the0 *Kepler* sample. The effective temperature thus seems to play a much larger role in the classification process than the different FliPer parameters. We decided to show classification results when removing the effective temperature from the input parameters in order to explain to what extent the effective temperature is needed for the classification.

When training and testing the classifier on *Kepler* data without using the effective temperature as an input parameter, the classification only depends on the FliPer parameters. With this configuration we obtain a 96% accuracy on the classification of the test set. Solar-like stars are still very well classified, and most errors concern the SPB/β-Cephei class (see Tab. 4). This is explained by Fig 1 where when noticing that $F_{p,i}$ ($i \in [0.7, 7, 20, 50]$ μHz) values for SPB/β-Cephei are quite similar to those of Solar-like stars and δ-Scuti/RoAp/sdBV.

With this study, we point out that using only the FliPer parameters as inputs to the algorithm is enough to recognise all Solar-like stars. However, adding physical parameters (such as $T_{\rm eff}$) to the classifier allows $\text{FliPer}_{\text{Class}}$ to perform well for all pulsators, and also to avoid false detection of solar-like stars as shown by Tab. 3.

5.2. Taking into account uncertainties on input parameters

In order to test the robustness of the classifier regarding uncertainties on the input parameters, we test the algorithm on the *Kepler* dataset with modified values of input parameters. Uncertainties on FliPer arise from the photon noise in the spectra (following a chi-squared distribution with two degrees of freedom). Hence, the uncertainty on the FliPer (see Bugnet et al. 2018a, for more details) parameters can be explicitly written as

$$\delta F_p = \sqrt{\overline{\delta PSD}^2} = \frac{\delta P_{tot}}{N_{bin}} . \quad (2)$$

We use the central limit theorem and we re-bin the spectrum by a factor of $n = 50$. The total amount of power in

the spectrum is

$$\mathrm{P}_{tot} = \sum_j \mathrm{P}_{\mathrm{n,j}} \ , \tag{3}$$

where $\mathrm{P}_{\mathrm{n,j}}$ follows a quasi-normal distribution with 2n degrees of freedom. It assumes that the signal does not change dramatically over this range of 50 bins, which is a large assumption for classical pulsators. It leads to a global uncertainty on FliPer values of

$$\delta \mathrm{F_p} = \frac{\sqrt{\sum_j \left(2 \times \frac{\mathrm{P}_{50,\mathrm{j}}}{2n} \times \sqrt{n}\right)^2}}{\mathrm{N_{bin}}} \ . \tag{4}$$

The effective temperature values for the *Kepler* set are taken directly from the Mathur et al. (2017) catalog. As long as no spectroscopic follow up surveys are available, only the effective temperature coming from the TIC will be available for most TESS data. Large uncertainties are expected, as on average $\overline{\delta T_{\rm eff}} \sim 170$ K according to the first sectors data. To be representative of future TESS data, we decide to use $\delta T_{\rm eff} = 170$ K instead of using uncertainties from the Mathur et al. (2017) catalog for the whole *Kepler* test sample.

We then include the effect of these errors on the different parameters during the testing of the algorithm. We perform a Monte Carlo simulation by generating for each star in our test sample 100 artificial sets of parameters from their corresponding normal distributions. We compute for each X parameter ($\mathrm{F}_{\mathrm{p}_{0.7}}$, $\mathrm{F}_{\mathrm{p}_7}$, $\mathrm{F}_{\mathrm{p}_{20}}$, $\mathrm{F}_{\mathrm{p}_{50}}$, and $T_{\rm eff}$) 100 new values $\mathrm{X}_{0 \leq i \leq 100}$ following

$$\mathrm{X}_{0 \leq i \leq 100} = \mathrm{X} + \delta \mathrm{X} \times \mathcal{G}_{0 \leq i \leq 100} \ , \tag{5}$$

with $\mathcal{G}_{0 \leq i \leq 100}$ being 100 random values following the standard distribution. Each new group of X_i parameters describes a new star to test the algorithm.

We keep training the algorithm by using the original *Kepler* training set. The new test set now contains a hundred times more stars than the original test set in order to include the effect of uncertainties on the input parameters. We are able to classify these new stars with a 99% accuracy. We thus conclude that there is no impact of uncertainties of the chosen input parameters for the classification of stars. This comes from the fact classes are well separated in the $\log(T_{\rm eff})$ vs. $\log(\mathrm{F}_{\mathrm{p},0.7})$ (see Fig. 2). In particular, large uncertainties on $T_{\rm eff}$ that are representative of future TIC effective temperatures do not perturb the classification of pulsators.

6. Conclusion

The study on *Kepler* data confirms the results obtained by using TESS simulated data. As expected, FliPer$_{\rm Class}$ is a great method to recognise solar-like stars from the shape of their PSD. By using $\mathrm{F}_{\mathrm{p},i}$ ($i \in [0.7, 7, 20, 50]$ μHz) along with $T_{\rm eff}$ as input parameters in a Random Forest algorithm, more than 98% of TESS simulated and almost the totality of the *Kepler* solar-like pulsators within the test set are well-classified amongst other pulsators. We plan to improve the $\mathrm{F}_{\mathrm{p},i}$ calculation (especially for stars observed by TESS with a low signal to noise ratio) by empirically calibrating the photon noise as a function of the TESS magnitude of the star (similar to the study by Jenkins et al. 2010, for the *Kepler* data) instead of measuring the power at high frequencies that can be biased by astrophysical signal. By comparing the results on noisy data with previous results obtained by using clean simulated data (Bugnet et al. 2018b) we notice that the performance of FliPer$_{\rm Class}$ is only slightly diminished by the presence of photometric noise. This is also auspicious for the applicability of the method to real TESS data. This study will help the massive seismic analysis of TESS solar-like stars with global seismic pipelines by providing a list of stars that are predicted to be solar-like stars.

FliPer$_{\rm Class}$ gives a large weight to seismology throught the use of the $\mathrm{F}_{\mathrm{p},i}$ parameters. We chose not to incorporate any *Gaia* parameters in the FliPer$_{\rm Class}$ to remain as general as possible. For example, for faint stars as the *Kepler* red giants at the deep end of the Milky Way (Mathur et al. 2016a) or for polluted systems, *Gaia* luminosities could have large uncertainties or even be biased. Hence, seismic parameters coupled to effective temperature could be a better choice as showed by Huber et al. (2017). Therefore, the FliPer$_{\rm Class}$ as defined here could be complemented by any additional precise astrometric, photometric or spectroscopic parameters, that could then be applied to any observations from *Kepler*, K2 or TESS missions.

FliPer parameters are integrated as features in the TASOC/T'DA Random Forest classifier that will be used to automatically classify all TESS targets. This enlarged Random Forest is itself part of a larger classifier which includes convolutional neural networks (Hon et al. 2018), clustering, etc. The pipeline (Tkatchenko et al., *in prep*) is currently being built to be efficient in classifying all types of pulsators, and should demonstrate a high level of performance even for stars with complicated pulsation pattern.

Acknowledgements. We thank all the T'DA team for useful comments and discussions, in particular Dr. Andrew Tkachenko. We also acknowledge Marc Hon, Keaton Bell and James Kuszlewicz for useful comments on the manuscript. L.B. and R.A.G. acknowledge the support from PLATO and GOLF CNES grants. S.M. acknowledges support by the Ramon y Cajal fellowship number RYC-2015-17697. O.J.H. and B.M.R. acknowledge the support of the UK Science and Technology Facilities Council (STFC). M.N.L. acknowledges the support of the ESA PRODEX programme (PEA 4000119301). Funding for the Stellar Astrophysics Centre is provided by the Danish National Research Foundation (Grant DNRF106).

References

Aigrain, S., Parviainen, H., & Pope, B. J. S. 2016, MNRAS, 459, 2408
Bai, L., Guo, P., & Hu, Z.-Y. 2005, Chinese J. Astron. Astrophys., 5, 203
Balona, L. A. 2013, MNRAS, 436, 1415
Balona, L. A., Cunha, M. S., Kurtz, D. W., et al. 2011, MNRAS, 410, 517
Breiman, L. 2001, Machine Learning, 45, 5
Bugnet, L., Garcia, R. A., Davies, G. R., Mathur, S., & Corsaro, E. 2017
Bugnet, L., García, R. A., Davies, G. R., et al. 2018a, A&A, 620, A38
Bugnet, L., Garcia, R. A., Davies, G. R., et al. 2018b, in SF2A-2018: Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics

Campante, T. L., Schofield, M., Kuszlewicz, J. S., et al. 2016, ApJ, 830, 138
Debosscher, J., Sarro, L. M., López, M., et al. 2009, A&A, 506, 519
Gandolfi, D., Barragan, O., Livingston, J., et al. 2018, ArXiv e-prints, arXiv:1809.07573
García, R. A., Hekker, S., Stello, D., et al. 2011, MNRAS, 414, L6
Goldreich, P. & Keeley, D. A. 1977, APJ, 212, 243
Hon, M., Stello, D., & Yu, J. 2018, MNRAS
Huang, C. X., Burt, J., Vanderburg, A., et al. 2018, ArXiv e-prints, arXiv:1809.05967
Huber, D., Stello, D., Bedding, T. R., et al. 2009, Communications in Asteroseismology, 160, 74
Huber, D., Zinn, J., Bojsen-Hansen, M., et al. 2017, ApJ, 844, 102
Jenkins, J. M., Caldwell, D. A., Chandrasekaran, H., et al. 2010, ApJ, 713, L120
Kallinger, T., Hekker, S., Garcia, R. A., Huber, D., & Matthews, J. M. 2016, Science Advances, 2, 1500654
Lebreton, Y. & Goupil, M. J. 2014, A&A, 569, A21
Li, G., Bedding, T. R., Murphy, S. J., et al. 2019, MNRAS, 482, 1757
Lund, M. N., Handberg, R., Kjeldsen, H., Chaplin, W. J., & Christensen-Dalsgaard, J. 2017, in European Physical Journal Web of Conferences, Vol. 160, European Physical Journal Web of Conferences, 01005
Mathur, S., García, R. A., Huber, D., et al. 2016a, ApJ, 833, 294
Mathur, S., García, R. A., Huber, D., et al. 2016b, ApJ, 827, 50
Mathur, S., García, R. A., Régulo, C., et al. 2010, A&A, 511, A46
Mathur, S., Hekker, S., Trampedach, R., et al. 2011, ApJ, 741, 119
Mathur, S., Huber, D., Batalha, N. M., et al. 2017, The Astrophysical Journal Supplement Series, 229, 30
Mathur, S., Huber, D., Batalha, N. M., et al. 2017, ApJS, 229, 30
McNamara, B. J., Jackiewicz, J., & McKeever, J. 2012, AJ, 143, 101
Mitchell, M. 2011, Open Journal of Statistics, 1 No. 3, 205
Molnár, L., Plachy, E., Juhász, Á. L., & Rimoldini, L. 2018, A&A, 620, A127
Mosser, B. & Appourchaux, T. 2009, A&A, 508, 877
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Pérez-Ortiz, M. F., García-Varela, A., Quiroz, A. J., Sabogal, B. E., & Hernández, J. 2017, A&A, 605, A123
Reed, M. D., Baran, A. S., Telting, J. H., et al. 2018, Open Astronomy, 27, 157
Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2014, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 9143, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 20
Sachkov, M. 2014, in Putting A Stars into Context: Evolution, Environment, and Related Stars, ed. G. Mathys, E. R. Griffin, O. Kochukhov, R. Monier, & G. M. Wahlgren, 315–322
Serenelli, A., Johnson, J., Huber, D., et al. 2017, The Astrophysical Journal Supplement Series, 233, 23
Smalley, B., Niemczura, E., Murphy, S. J., et al. 2015, MNRAS, 452, 3334
Stassun, K. G., Oelkers, R. J., Pepper, J., et al. 2018, AJ, 156, 102
Veljanoski, J., Helmi, A., Breddels, M., & Posti, L. 2018, ArXiv e-prints, arXiv:1804.05245
Viquar, M., Basak, S., Dasgupta, A., Agrawal, S., & Saha, S. 2018, ArXiv e-prints, arXiv:1804.05051

Appendix A: Random Forest Classifier

Appendix A.1: Supervised machine learning

Random Forests are supervised machine learning (ML) algorithms, which learn how to predict an output variable ($\mathrm{Y_{predicted}}$) from some training data (X) for which the corresponding result ($\mathrm{Y_{known}}$) is already known. It learns a mapping function f from the input(s) to the output:

$$\mathrm{Y_{predicted}} = f(X) \tag{A.1}$$

The algorithm iteratively makes predictions ($\mathrm{Y_{predicted}}$) on the training data (X). They are corrected to achieves a maximum level of performance, by comparing with the $\mathrm{Y_{known}}$ classes. The out-of-bag (OOB) error evaluates at each step the performance of the algorithm. We use a surpervised ML algorithm for our study because we have input variables X (which are $\mathrm{F_{p_{0.7}}}$, $\mathrm{F_{p_7}}$, $\mathrm{F_{p_{20}}}$, $\mathrm{F_{p_{50}}}$, and $T_{\rm eff}$) and an output class $\mathrm{Y_{known}}$ (representing the type of pulsator).

Appendix A.2: Classification trees

The clasification tree method is part of the Classification and Regression Trees (CART) introduced by Breiman (2001). A decision tree algorithm construct a binary tree during the training, with each node representing and a split point on an input variable (X) (numerical value for regression algorithms, or class name for classification algorithms). The leaf nodes of the tree contain the output possible classes ($\mathrm{Y_{predicted}}$).

The tree is build such as a cost function is minimized at each node. Equation A.2 is the cost function used for the classifier, with $\mathrm{N_{classes}}$ the number of classes and p_k the number of training instances with class k at the node of interest.

$$\mathrm{G} = \sum_{k=1}^{\mathrm{N_{classes}}} p_k \times (1 - p_k) \tag{A.2}$$

Once the tree is build on the training sample, it is used to evaluate $\mathrm{Y_{predicted}}$ for new $\mathrm{X_{new}}$ data.

Appendix A.3: Ensemble method Random Forest classifier

An ensemble method combines the prediction from multiple ML algorithms together. It aims at making even more accurate predictions than any individual model. The Random Forest classifier is an ensemble method that combines classfication trees. It consists in:

- Creating many sub-samples of the training sample.
- Training a classification tree on each sub-sample, with keeping a low number of features that can be looked at for each split point. It aims at decreasing the correlation between the different trees. For classification algorithms the maximal number of features searched at each split point is usually $m = \sqrt{p}$, with p the number of input (X) variables.
- Calculating the dominant class from each model for the new test sample: this predicted class is used as the output variable ($\mathrm{Y_{predicted}}$).