Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **497,** 4843–4856 (2020)
Advance Access publication 2020 July 27

doi:10.1093/mnras/staa2155

Clumpiness: time-domain classification of red giant evolutionary states

James S. Kuszlewicz [1,2]★ Saskia Hekker [1,2] and Keaton J. Bell [3]†

[1]*Max-Planck-Institut für Sonnensystemforschung, Justus-von-Liebig-Weg 3, D-37077 Göttingen, Germany*
[2]*Stellar Astrophysics Centre, Department of Physics and Astronomy, Aarhus University, Ny Munkegade 120, DK-8000 Aarhus C, Denmark*
[3]*DIRAC Institute, Department of Astronomy, University of Washington, Seattle, WA 98195, USA*

Accepted 2020 July 9. Received 2020 July 8; in original form 2020 April 27

ABSTRACT

Long, high-quality time-series data provided by previous space missions such as *CoRoT* and *Kepler* have made it possible to derive the evolutionary state of red giant stars, i.e. whether the stars are hydrogen-shell burning around an inert helium core or helium-core burning, from their individual oscillation modes. We utilize data from the *Kepler* mission to develop a tool to classify the evolutionary state for the large number of stars being observed in the current era of K2, *TESS*, and for the future *PLATO* mission. These missions provide new challenges for evolutionary state classification given the large number of stars being observed and the shorter observing duration of the data. We propose a new method, Clumpiness, based upon a supervised classification scheme that uses 'summary statistics' of the time series, combined with distance information from the *Gaia* mission to predict the evolutionary state. Applying this to red giants in the APOKASC catalogue, we obtain a classification accuracy of $\sim$ 91 per cent for the full 4 yr of *Kepler* data, for those stars that are either only hydrogen-shell burning or also helium-core burning. We also applied the method to shorter *Kepler* data sets, mimicking *CoRoT*, K2, and *TESS* achieving an accuracy $>$ 91 per cent even for the 27 d time series. This work paves the way towards fast, reliable classification of vast amounts of relatively short-time-span data with a few, well-engineered features.

Key words: asteroseismology – methods: data analysis – methods: statistical.

1 INTRODUCTION

Stars in the red clump (RC), i.e. low-mass core-helium-burning (CHeB) stars, are a prominent feature in the colour–magnitude diagram of, for instance, open or globular clusters due to their narrow luminosity distribution. This narrow distribution is a result of the near-constant core mass at helium ignition of stars with degenerate cores. RC stars are widespread throughout the Galaxy and are used as distance indicators (e.g. Girardi et al. 1998; Paczyński & Stanek 1998; Groenewegen 2008). Furthermore, their function as standard candles provides the possibility to use the parallaxes from the *Gaia* mission (Gaia Collaboration 2016, 2018) to calibrate the absolute magnitude of the RC (e.g. Hawkins et al. 2017; Chan & Bovy 2020; Hall et al. 2019). The RC has also been used as a benchmark to investigate possible systematics in the parallax measurements when combined with, for example, asteroseismology (e.g. Davies et al. 2017; Huber et al. 2017; Khan et al. 2019; Zinn et al. 2019).

The identification of field RC stars is not trivial because they have very similar surface properties to red giant branch (RGB) stars. To add to this, not all field stars are at the same distance and so the typical 'clump' seen in the colour–magnitude diagram of clusters is smeared out. There are also additional dependences on parameters such as extinction and metallicity (e.g. Girardi & Salaris 2001). Bovy et al. (2014) identified RC stars in APOGEE data by not only searching luminosity space, but also colour–metallicity-surface-gravity-effective temperature space in order to reduce the amount of contamination from RGB stars.

With the advent of *CoRoT* (Baglin et al. 2006) and *Kepler* (Borucki et al. 2010), the evolutionary state of red giant stars can now also be classified through asteroseismology (e.g. Bedding et al. 2011; Mosser et al. 2011; Kallinger et al. 2012; Stello et al. 2013; Mosser et al. 2014; Vrard, Mosser & Samadi 2016; Elsworth et al. 2017; Hekker et al. 2017; Hon, Stello & Yu 2018). The majority of these methods rely on the detection of individual mixed oscillation modes (those that have a gravity-mode-like character in the stellar core and pressure-mode-like character in the convective envelope) in the power spectrum. These modes are indeed resolved for 4 yr of *Kepler* data (hereby defined as long data sets). However, for the shorter 80-d data sets of K2 (Howell et al. 2014) or the 27-d time series of *TESS* (Ricker et al. 2015; hereby referred to as short data sets), the frequency resolution is coarser, hampering the evolutionary state classification via mixed modes. To distinguish between RC and RGB stars, it is therefore important to have classification methods that do not rely on the determination of individual mixed oscillation modes a priori. This will also be of importance for future space missions such as *PLATO* (Rauer et al. 2014).

Evolutionary state determination does not have to be undertaken in the frequency domain and we show in this work how the time-series data can be used directly. In this way, we are not limited by the reduced frequency resolution in the power spectra for shorter time-series data.

★ E-mail: kuszlewicz@mps.mpg.de
† NSF Astronomy and Astrophysics Postdoctoral Fellow and DIRAC Fellow.

It is common practice to classify the variability of stellar light curves in the time domain using machine-learning techniques. Unsupervised techniques (those where the stellar classification is not known beforehand) have been applied to intrinsically variable stars (e.g. Cepheids/RR Lyraes, delta-Scuti pulsators etc.) and extrinsically variable stars (e.g. eclipsing binaries) in K2 data by Armstrong et al. (2015, 2016), who used a combination of self-organizing maps (to extract salient features from the data) and random forests to classify the data. Valenzuela & Pichara (2018) used a different approach to classify variable stars, applied to data from OGLE (Udalski et al. 1996, 2008), MACHO (Alcock et al. 1997, 2003), and *Kepler*, based upon the similarity between different time series using a variability tree, which ranks light curves by their similarity. Naul et al. (2018) took a different approach still and successfully used a 'novel' recurrent autoencoder structure to classify variable stars with irregularly sampled data from ASAS (Pojmanski 2002). The advantage of such a method is that no features are chosen a priori and the autoencoder instead extracted them automatically. This method was only optimized for periodic variables. Kügler, Gianniotis & Polsterer (2015) chose yet another method and adopted featureless classification by decomposing data into a mixture of Gaussians and using a distance matrix to classify similar targets, with applications to OGLE and ASAS. Many of the cases mentioned above used short or sparsely sampled data sets, whereas in the case of *Kepler* data the sheer amount of data collected can be a bottleneck in the extraction of important features. For many types of intrinsically variable stars, the periods of oscillation dictate that densely sampled time series is necessary to resolve the underlying signal.

In this work, we introduce Clumpiness, an approach aimed at deducing the evolutionary state of red giants in the time domain, while retaining interpretability and computational efficiency to create a classification tool that performs as well as possible on both long and short time-series data using the fewest number of features.

2 FEATURES

A key component of any machine -learning scheme is the set of features that are extracted from the data. The features form the backbone of the analysis and are what the chosen algorithm will use to predict the class an object belongs to. In our case, this is the evolutionary state of a red giant star.

There are a number of packages in the literature for computing very general features for time-series classification, for example, TSFRESH (Christ, Kempa-Liehr & Feindt 2016), FATS (Feature Analysis for Time Series; Nun et al. 2015), UPSILON (Kim & Bailer-Jones 2015), and PTSA[1] to name but a few. Rather than computing a large number of very general features, we instead compute a small number of features in this work that contain relevant information about the evolutionary state of the stars in question.

2.1 Time-series features

In this work, we focused on red giants that show solar-like oscillations that are excited and damped by near-surface turbulent convection. The near-surface turbulent convection displays itself as granulation that is present in the power spectrum as a frequency-dependent signal (commonly modelled as red noise). Whereas the solar-like oscillations are only present in the power spectrum at specific frequencies, as defined by the underlying stellar properties. The dominant process contributing approximately 85 per cent of the variability power in the time series is the granulation, determined by integrating separately the granulation and oscillations components of models fit to the power spectrum. However, both the granulation and oscillation signals can provide insights into the evolutionary state of the star.

2.1.1 *Time-series MAD*

The variance of time-series data of a star with a convective envelope containing granulation and solar-like oscillations scales with $\nu_{\rm max}$ (the frequency of maximum oscillation power, Hekker et al. 2012), which in turn scales with the stellar surface gravity (e.g. Brown et al. 1991; Kjeldsen & Bedding 1995).

To capture a measure of the variance, we compute the MAD (median absolute deviation from the median)

$$\mathrm{MAD}(X_0) = \mathrm{median}\left(\left|X_{0,i} - \mathrm{median}(X_0)\right|\right), \quad (1)$$

where X_0 represents the whole time series, and $X_{0,i}$ is a single data point from that time series. We prefer the MAD over the variance as a feature because it is more resistant to outliers (e.g. Leys et al. 2013).[2]

2.1.2 *MAD of time-series first differences*

Alongside computing the MAD of the time series, we also look at the MAD of the first differences, where we define the first differences X_1 of a time-series X_0 to be

$$X_{1,i} = X_{0,i} - X_{0,i-1}. \quad (2)$$

The first differences remove any long term trends from the data revealing additional time-scales in the data. The MAD (equation 1) of the first differences provides information regarding the magnitude of the rate of change of the signal. Therefore, stars with longer period granulation will have lower first-difference MAD values compared to those with shorter period granulation. The MAD of the first differences acts as a rudimentary high-pass filter and removes power from the low-frequency regime of the signal. This will, in general, suppress the contribution of the granulation to the signal and enhance the contribution from the oscillations (and possibly the white noise if the level is very high). This offers a different view of the data that is complementary with taking the MAD of the time series.

2.1.3 *Normalized number of zero-crossings*

To capture the dominant time-scale, we compute the normalized (by the number of data points) number of zero-crossings in the time-series data. This normalization assumes that the time stamps where there are no data points (i.e. during gaps) are dropped and so the number of data points used to normalize the data are the actual number of data points irrespective of their distribution in time, i.e. in the case of gaps. This normalization is to place all stars on to the same scale enabling direct comparisons between data sets of different length. We also correct the normalized number of zero-crossings for gaps in the data if needed (see Appendix A for more information).

[1] https://github.com/compmem/ptsa

[2] Note that the MAD can be converted to the variance by assuming that the data are Gaussian-distributed, which may not always be the case.

We compute the zero-crossings by first constructing a 'clipped' time series, Z_0, from the original time-series X_0 of length N such that

$$Z_{0,i} = \begin{cases} 1 & \text{if } X_{0,i} \geq 0, \\ 0 & \text{if } X_{0,i} < 0. \end{cases} \tag{3}$$

The number of zero-crossings, D_0, is then given by the number of value changes in the new 'clipped' time series, computed as

$$D_0 = \sum_{i=1}^{N-1} \left(Z_{0,i} - Z_{0,i-1}\right)^2, \tag{4}$$

where $0 \leq D_0 \leq N-1$.

The total number of zero-crossings are normalized to obtain the relative number of zero-crossings for a time series of a given length, as described in Appendix A.

2.1.4 *Signal coherency*

We can expand on the idea of the number of zero-crossings in the time series by going to higher order differences, i.e. first differences of the time series (equation 2), first differences of the first differences, and so on. We compute the 'clipped' time series of the higher order crossings (HOCs) in exactly the same way as for the original time-series data, as given by equation (4), except now we use the time series of higher order differences as an input. The time series of the kth HOCs is defined by

$$X_{k,i} = X_{k-1,i} - X_{k-1,i-1}. \tag{5}$$

Denoting the original 'clipped' time series as Z_0, we define the 'clipped' time series of the kth HOC time series by Z_k such that

$$Z_{k,i} = \begin{cases} 1 & \text{if } X_{k,i} \geq 0, \\ 0 & \text{if } X_{k,i} < 0. \end{cases} \tag{6}$$

The number of HOCs becomes

$$D_k = \sum_{i=1}^{N-1} (Z_{k,i} - Z_{k,i-1})^2. \tag{7}$$

We combine the information contained within the number of HOCs to maximize the information content of this set of features. Solar-like oscillations and granulation are stochastically driven with lifetimes/time-scales that change depending on the size of the star. For a fixed length of time-series data, the signal from a star that is more evolved (i.e. with longer granulation time-scales) will appear more coherent than that of a star that is less evolved. In contrast to the more coherent signal as a star evolves, the incoherent nature of a white noise signal forms the lower boundary of the coherency measure. Therefore, we will make use of the HOCs to compute a measure of the coherency of the time series.

To compute a measure of the coherency, we look at the increments of the HOCs (Bae et al. 1996), i.e. the rate of change of the HOCs as a function of their order. These are defined as

$$\Delta_k = \begin{cases} D_0 & \text{if } k = 0, \\ D_k - D_{k-1} & \text{if } k = 1, ..., N-2, \\ (N-1) - D_{k-2} & \text{if } k = N-1. \end{cases} \tag{8}$$

Kedem & Slud (1981) have shown that for $N \geq 300$ (where N is the number of data points in the time series), the number of HOCs increases for $k \leq 7$. We can therefore assuming that our HOCs are ordered such that the increments by which they increase are always positive. Kedem & Slud (1982) also showed that when $k = 8$, D_k/N (the high-order crossings normalized by the number of data points N) is already as large as 0.9. As a result, the discriminatory power is greatly reduced when k is larger because D_k/N levels off and slowly approaches unity. In this work, we found that $k = 5$ is an acceptable upper limit that balances discriminatory power with computation time.

The increments are combined into the coherency measure ψ^2

$$\psi^2 = \sum_{k=0}^{K-1} \frac{(\Delta_k - \phi_k)^2}{\phi_k}. \tag{9}$$

For $k = 0, ..., K$, where $K = 5$ the increments simulated for a white noise time series of length 100 000 points are given by $\phi_k = (0.167, 0.066, 0.038, 0.025, 0.018)$.

The coherency measure ψ^2 provides a way to discriminate between white noise-like signals and those that are more coherent. For instance, for a purely sinusoidal (coherent) signal the increments Δ_k will be zero for all k. This is because the period of the signal does not change when the differences are taken and so

$$\psi^2_{\rm sin} = \sum_{k=0}^{K} \phi_k. \tag{10}$$

For a completely incoherent signal (i.e. white noise), the increments are equal to ϕ_k because they are generated from the same underlying process and therefore

$$\psi^2_{\rm white} = 0. \tag{11}$$

Therefore, the upper and lower bounds of ψ^2 are defined such that $0 \leq \psi^2 \leq \sum_{k=0}^{K} \phi_k$. Note that we normalize ψ^2 in the same way as the number of zero-crossings (see Appendix A).

2.2 Absolute *K*-band magnitude

Alongside information gathered from the time series, we include external information that contributes to improving the classification. We mentioned in the introduction that the RC forms a well-defined region in luminosity and therefore absolute magnitude space, so leveraging this information is particularly helpful. We therefore use parallax data from the *Gaia* mission (Gaia Collaboration 2016, 2018). Rather than using the parallax as a feature, we link this back to an intrinsic property of the star, namely the absolute magnitude, that can help distinguish RGB from RC. We choose to compute the absolute magnitude in the K band because the contribution of interstellar extinction is greatly reduced and in most cases negligible. This reduces the effect of any possible sources of bias in the dust maps used to calculate the absolute magnitude accurately.

The absolute magnitude is calculated in the K band according to

$$M_{K_{\rm s}} = m_{K_{\rm s}} - \mu_0 - A_{K_{\rm s}}. \tag{12}$$

The apparent K-band magnitude of the star $m_{K_{\rm s}}$ is taken from the Two Micron All Sky Survey (2MASS; Cutri et al. 2003; Skrutskie et al. 2006). We compute the interstellar extinction in the K band, $A_{K_{\rm s}}$, from the 3D dust map of Green et al. (2015). The distance modulus, $\mu_0 = 5\log_{10}(d) - 5$ where the distance, d, (in parsecs) is taken from Bailer-Jones et al. (2018). We use a 4 arcsec cross-match between the asteroseismic and *Gaia* data sets, as given by gaia-kepler.fun, and take the brightest G-band magnitude source in the case of multiple *Gaia* sources for a single asteroseismic object. We adopt the following selection criterion (Lindegren et al. 2018) to

ensure that the stars that satisfy this have well-derived parallaxes:

$$\sqrt{\frac{\chi^2}{\nu}} < 1.2 \times \max\{1, \exp[-0.2\,(G - 19.5)]\}\,, \tag{13}$$

where χ^2 is the goodness-of-fit of the single-star astrometric model as provided by *Gaia*, ν is the number of degrees of freedom in the astrometric model, and G is the mean *Gaia* magnitude. If a star does not satisfy the above criterion then it is not included in the training data (see next section) for the full classification algorithm, but is instead included in the training of a version of the algorithm without M_{K_s} included as a feature.

3 TRAINING DATA

In this work, we use red giants from the APOKASC sample (Pinsonneault et al. 2018) observed by *Kepler* with long-cadence (29.42 min) time-series observations as training data. This is a sample of well-studied red giant stars that have consolidated evolutionary states from four different methods (Elsworth et al. 2019). We also include the sample of 472 main-sequence and subgiant stars from Chaplin et al. (2014) and a random sample of 1500 *Kepler* objects of interest (KOIs) from the NASA Exoplanet Archive (Akeson et al. 2013) with $\log g > 3.5$ in the training data. These stars are not expected to show solar-like oscillations in the frequency range associated with long-cadence observations but are common contaminants in red giant samples (e.g. Yu et al. 2016) and act here as a representation of those contaminants that could find their way into such samples.

3.1 Time-series preparation

All the time series used as training data are from long-cadence *Kepler* observations produced using the pipeline developed by Jenkins et al. (2010) with an additional Savitzky–Golay filter of width 20 d applied to remove any residual long-term trends. We use the LIGHTKURVE package (Lightkurve Collaboration 2018) for all pre-processing of the data. There are regular gaps in the time series caused by the desaturation of the reaction wheels on the *Kepler* spacecraft, typically the duration of one long-cadence (29.42 min) observation every 3 d. These were filled using linear interpolation according to García et al. (2014). The data are then concatenated if segments of the data are separated by a long gap (over 20 d) following Hekker et al. (2010).

Alongside the full 4 yr of data, we also produce reduced length data sets that correspond to the time base of other missions, such as 80 d for K2 (Howell et al. 2014), 180 d for *CoRoT* (Baglin et al. 2006), and 27 d for *TESS* (Ricker et al. 2015). For *TESS*, we consider only the shortest length data sets as the limiting case in terms of data set length.

To prepare shorter length data sets for algorithm training, we cut down the raw (now concatenated) data into chunks of the desired length (in time) and filter them with a Savitzky–Golay filter of width 10 d for the 180 d data sets, 5 d for 80 d data sets, and 3 d for 27 d data sets. These filters mimic the detrending that would be used for the shorter data sets. In the case of the shortest length of 27 d, a linear trend is subtracted after the filtering to account for any possible trends in the data as a result of the short baseline. If any chunk of data has a duty cycle lower than 0.5 (i.e. more than 50 per cent of data are missing), then the chunk is discarded. This rarely happens since the data are concatenated in the presence of large gaps and so in most cases the duty cycle is above 0.9. Due to the stochastic nature of solar-like oscillations, we are essentially gaining more independent training time series when cutting the longer time series down. This is due to the fact that the phase is incoherent in solar-like oscillators.

There is the possibility for a time series of a given star to be contaminated by a nearby star due to the mask chosen for the extraction of the data. We do not want contaminated time series to be used in our training set since this will confuse the classification algorithm. To remove contaminated time series, we use a metric based on the quarter-by-quarter variance, $\sigma^2_{q,i}$. We use the following metric on the filtered data (see above):

$$c = \text{median}\left(\sum_{i=2}^{N_{\text{quarters}}} \left|\sigma^2_{q,i} - \sigma^2_{q,i-1}\right|\right). \tag{14}$$

This is the median of the absolute values of the first differences of the quarter-by-quarter variances. When $\log_{10} c > 2.5$, which is determined empirically, the time series is considered contaminated and is therefore no longer used.

3.2 Class labels

The evolutionary states for the red giants that make up our ground truth labels are taken from the consolidated classifications of the APOKASC sample by Elsworth et al. (2019). Based on their classification and the inclusion of the main-sequence/KOI targets, we define three classes: RGB, CHeB stars, and main-sequence stars/KOIs (noise).

In order to aid the classifier and given that the RGB class stretches over a wide range of parameter space, we split up the RGB class (for training purposes) into three subclasses

(i) Low-luminosity RGB (LLRGB) with $\nu_{\max} > 130$ μHz.
(ii) High-luminosity RGB (HLRGB) with $\nu_{\max} < 15$ μHz.
(iii) Confusion region RGB (ConfRGB) with $15 < \nu_{\max} < 130$ μHz.

In this way, the classifier can focus on disentangling CHeB stars from RGB stars in the region where they overlap (the so-called confusion region). We do not disentangle the low-mass RC stars from the higher mass stars in the secondary clump. However, this could be performed afterwards using the probabilities from the classifier as an initial guide. Final probabilities are produced for the three main classes RGB, CHeB, and noise.

4 SUPERVISED CLASSIFICATION

The task of inferring the stellar evolutionary states from a set of derived features constitutes a supervised classification problem. In Fig. 1, we plot each feature against every other for our training set, coloured by class label. This shows that the interactions between our features are not necessarily linear. They may be linear in log-space or under a more complicated transformation. In order to take advantage of this, we use an ensemble algorithm `xgboost` (Chen & Guestrin 2016). An ensemble algorithm creates a sequence of weak classifiers whose individual performances are only slightly better than random guessing. When these weak classifiers are combined they produce a single strong classifier (e.g. Friedman, Tibshirani & Hastie 2000; Hastie, Tibshirani & Friedman 2001).

The `xgboost` algorithm is a variant of ensemble algorithms that falls under the umbrella of gradient-boosting methods. In boosting algorithms, models are constructed sequentially to correct the errors from the existing models until no further improvement can be made (according to some chosen criterion). In `xgboost`, these models are decision trees. A gradient-boosting algorithm is a boosting algorithm where a gradient-descent algorithm is used to minimize the objective function (e.g. negative log-likelihood) to train the model. At each

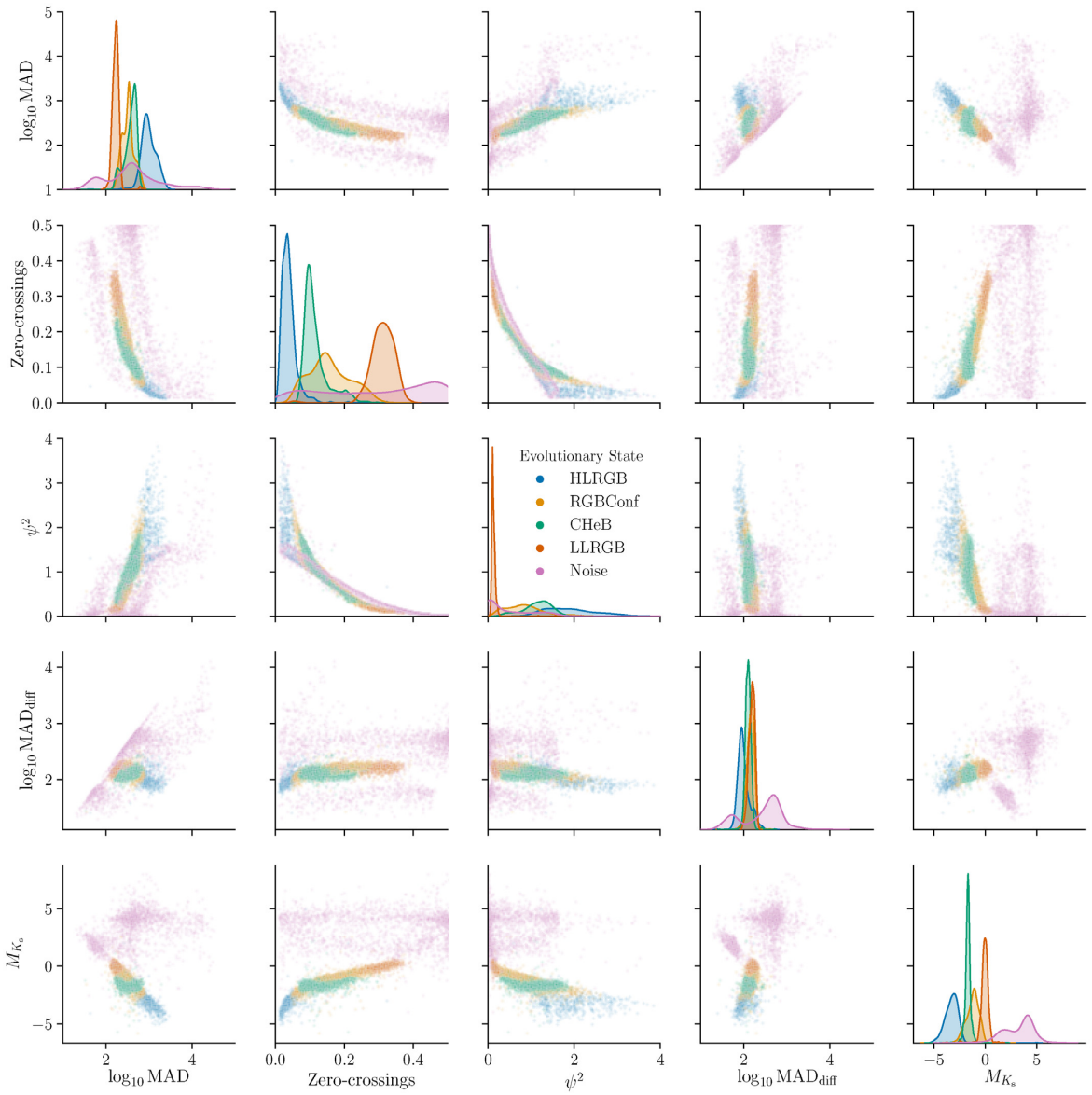

Figure 1. A pairplot showing kernel density estimates (KDE) of all of the features (Section 2) broken down into individual classes as shown on the diagonal: HLRGB, RGB stars that overlap with CHeB stars (RGBConf), CHeB, LLRGB, and the noise class. The relationships between each set of features are shown on the off-diagonal panels.

iteration, the data that are misclassified by the previous classifier in the sequence are upweighted such that the next classifier focuses more on those data.

In order to fit the model to the data during training, an objective function needs to be chosen. This is the function that we are minimizing via gradient descent in order to find the best fit to the data. We adopt the multiclass logarithmic loss since our problem is multiclass classification. This essentially acts as a measure of the degree of similarity between the ground truth and the predicted class probability, which is defined by

$$\ln \mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \ln p_{ij} + \sum_{k} \Omega(f_k). \tag{15}$$

The double summation is over each of the N time series in our training set and M classes, y_{ij} is a binary indicator showing whether the label j is the correct classification of the ith data point, and p_{ij} is the probability of the ith data point belonging to the jth class. $\Omega(f_k)$ is an extra term specific to `xgboost` which acts as a regularization term for the kth tree (f_k) used to construct the model (see Chen & Guestrin 2016, for more details) which essentially penalizes the size of the tree.

There are a number of hyperparameters in `xgboost` that can affect the performance of the classifier and need to be optimized, such as the size of each tree. This is performed using the HYPEROPT package (Bergstra, Yamins & Cox 2013) (see Appendix B for more details). Once the hyperparameter values are chosen, the classifier is trained according to a 10-fold stratified cross-validation scheme to choose the optimal number of trees (up to a limit of 1000). The stratification of the cross-validation folds ensures that there is the same relative number of stars in each class in the training and validation sets. At each step, a new tree is added to the model and

Table 1. Classification accuracy of the validation set for each time-series length with and without the inclusion of M_{K_s} as a feature.

Length of data set	Accuracy (per cent)	Accuracy (per cent) without M_{K_s}
4 yr	92.6	87.2
180 d	91.2	87.1
80 d	91.2	87.2
27 d	91.3	86.5

the loss function is evaluated on both a training set and validation set defined by the cross-validation fold. The training is stopped when the loss of validation set no longer decreases and starts to increase (signifying overfitting). The optimally trained model is taken as the last iteration with a decreasing validation set loss.

5 RESULTS

We train the classification algorithm on the data sets of different lengths and assess the accuracy (i.e. the percentage of stars correctly classified) and reliability of each case. The classifier produces a probability that the star belongs to each class. It is common to select the final class labels as the class with the largest probability, which is the same as choosing a probability threshold of $1/C$ where C is the number of classes.[3] However, in this work we opt to tune the threshold to produce the largest number of true positives and lower number of false positives for the trained classifier. The details of this are explained later in this section. The classification accuracy of the classifier for each data set length is shown in Table 1. We achieve greater than 91 per cent accuracy across all the different lengths of time series. The quoted accuracies take into account the three classes we want to classify the data into.

It is important not to disregard the class probabilities as they can provide useful information about the star being classified that is not apparent from the label alone. If the class probabilities are very similar then this means that the star could reasonably belong to any class, which would not be known if only the label is taken into account. In addition, the probabilities themselves can be used in subsequent probabilistic analyses as priors on the evolutionary state, for example in the classification effort being undertaken by the *TESS* Asteroseismic Consortium (TASC; Tkachenko et al., in preparation) or in studies of the RC (e.g. Hall et al. 2019; Khan et al. 2019).

5.1 Model evaluation

The performance of the trained models can also be assessed using the receiver-operator characteristic (ROC) curve that shows the diagnostic ability of a classifier. The curve is built up by computing the number of true and false positives in a given class for a number of different probability thresholds (decreasing from 1 to 0). The true positive rate is plotted against the false positive rate which gives us the ability to see how well the classifier correctly predicts the class for different thresholds. The ROC curves are shown for each class (using a one versus rest methodology, e.g. Bishop 2006) and each data set length in Fig. 2. The probability threshold for a star to belong to a certain class is chosen using the threshold that maximizes Youden's J statistic (defined by the difference between the true positive rate and the false positive rate for a given threshold; Youden 1950). The probability thresholds are therefore different for each class, reflecting the differing ability of the classifier to infer certain classes. The probability thresholds used are given in Table 2 and indicated by the coloured points in Fig. 2.

Ideally, the ROC curve should be as close to the top left corner as possible, showing that there is a threshold for which the classifier makes overwhelmingly accurate predictions with very few false positives. In our case, the ROC curves show that for the Noise class the curve is very close to the ideal top left corner and so for each time-series length the classifier is able to infer this class very well, with a very low number of false positives. For the CHeB and RGB classes, the performance is also very good, although we can see the interaction between the two classes manifesting itself as a reduction in the increase of the true positive rate above a value of 0.8. This provides important information about the false positive rates for the two classes. The fact that the CHeB class true positive rate increases faster than the RGB class shows that the CHeB class contains RGB false positives (as can be seen in Fig. 3), which is something we expect given the overlapping of the classes in feature space. While the RGB class will also contain CHeB false positives there are far fewer of these cases.

The area-under-the-curve (AUC) of the ROC curve gives the probability that the classifier will assign a higher probability to a random sample drawn from a chosen class (i.e. the class the AUC is computed for) compared to a random sample from a different class. The higher the AUC, the better the classifier is at predicting the correct class. For all of our classifiers these values, given in the legends of Fig. 2, are close and show that the overall classification for all the classes is very good. The confusion region where the RGB stars and CHeB stars overlap is what causes the AUC values for the RGB and CHeB classes to be slightly lower than the AUC value for the Noise class.

In addition to the accuracy of the trained classification models, we also look at the confusion matrices to gain a sense of which classes the classifier predicts correctly and which provide the most difficulty, as shown in Fig. 3. The diagonal of the confusion matrix shows the proportion of stars in a given class that have been correctly classified and the off-diagonal elements show how a star of a given class has been misclassified. Looking at Fig. 3 it is apparent that the main source of confusion is between the CHeB class and the RGB class. This is expected as in the confusion region the two classes overlap that will cause confusion and results in a drop in accuracy.

5.2 Model intepretation

The interpretability of the trained model is key to understanding how it performed and, more importantly, why it is performing in such a way. We use the feature importance of the classifier to interpret the model and which features it uses to classify stars belonging to different classes. We choose to use SHapely Additive exPlanation (SHAP) values (see Lundberg & Lee 2017; Lundberg, Erion & Lee 2018; for a more in-depth overview) to provide feature importances. The feature importance information is presented in Fig. 4 as a 'feature matrix', which contains the SHAP value averaged over each star in a given class normalized such that the rows sum to 1. Each row of the matrix then tells us the feature importance for the class represented by that row, and the column shows us the relative feature importance between each class.

We do not necessarily expect the same ranking of features for each length of data set since differences in the data set length may have an effect on the ability to derive given features. For example, the shorter the data set the more likely the variance is to differ from realization

[3]If more than one class exceeds the chosen threshold, we assign the class with the highest probability.

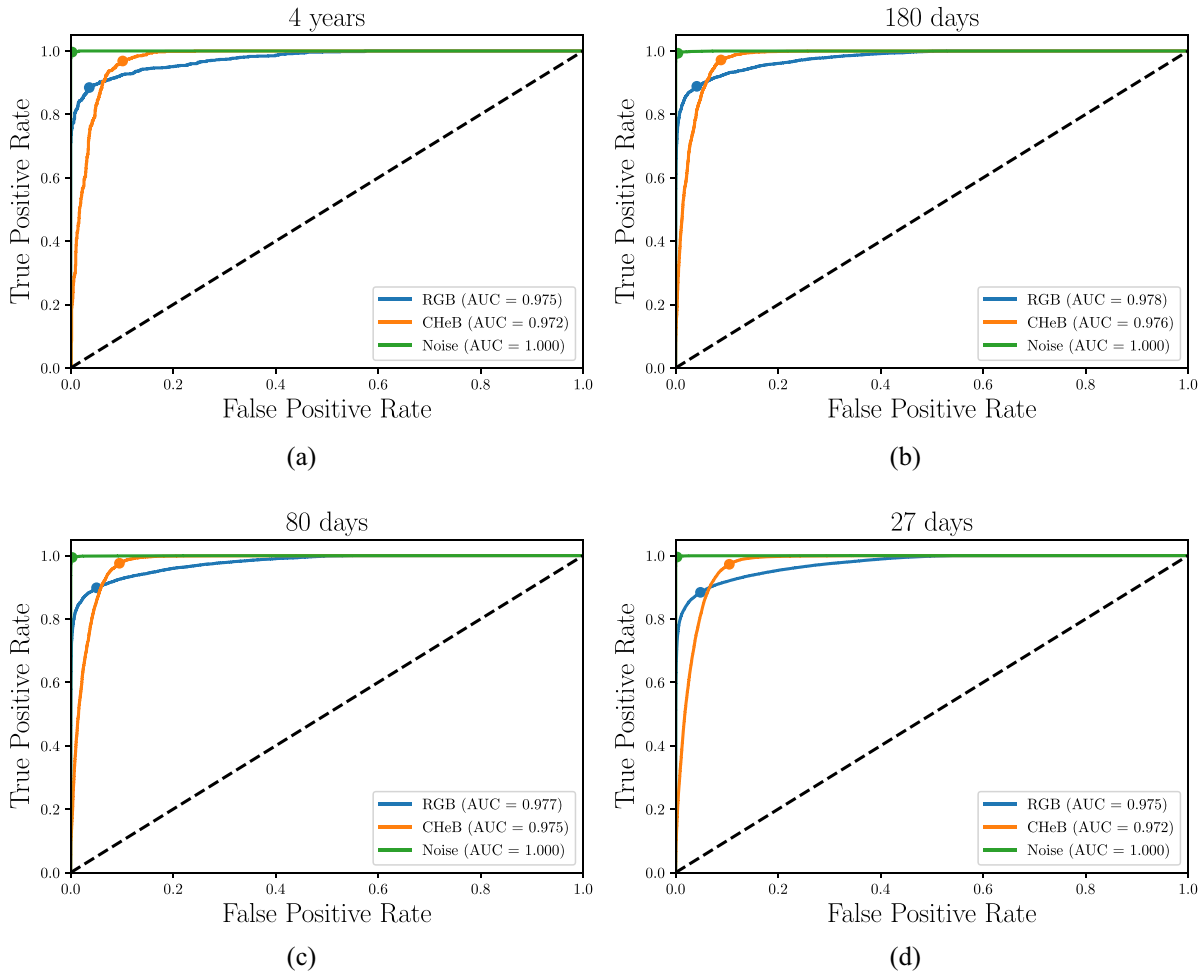

Figure 2. ROC curves for each of the different time-series lengths. The AUC (see text) for each curve is indicated in the legend and the coloured dots show the false positive and true positive rates for the chosen probability threshold. The dashed line indicates the 1-to-1 relation.

Table 2. Probability thresholds for a given class as a function of data set length.

Length of data set	Class RGB	RC	'Noise'
4 yr	0.385	0.493	0.427
180 d	0.395	0.283	0.378
80 d	0.420	0.307	0.102
27 d	0.393	0.349	0.145

to realization due to the stochastic nature of the data. Across the different lengths of data, M_{K_s} is the most important feature for the majority of classes, which is to be expected. Whereas for the LLRGB stars, the most important feature is ψ^2 because this provides the best way to distinguish between the nearby Noise class. Therefore, it is important to consider a local treatment of feature importance, otherwise a feature may be disregarded when looking globally when in fact it is vital to the accurate classification of a specific class.

6 APPLICATION TO THE FULL *KEPLER* LONG-CADENCE DATA SET

The subsample of data that we are looking at in this work, red giants showing solar-like oscillations, may not give us a full insight into feature space. Since all of our stars are stochastic oscillators, the underlying process generating the oscillations is the same. In order to interpret the features fully it is helpful to expand the region of parameter space with more stars with different types of intrinsic variability (e.g. activity and stellar oscillations) and extrinsic variability (e.g. eclipsing binaries). This can help identify contaminants in future samples. For instance, a star that has more than one type of variability, e.g. solar-like oscillations and eclipses, or this can be used to classify a wider range of stellar variability types. To help with our interpretation, we opt to extract features from every object observed in long cadence with *Kepler*. Since we will now have many different types of objects, e.g. eclipsing binaries, classical pulsators etc., this will not only enable us to improve the interpretation of our features, but also show the viability of using the derived features to discern other variable classes of star.

Let us focus on two parameters in particular, the number of normalized zero-crossings and the coherency parameter ψ^2. These two parameters provide information regarding the time-scale of the dominant contribution to the light curve and the degree of coherency (or stochasticity). Fig. 5 shows these features for the full *Kepler* sample in a reduced region of parameter space, which we shall explain before moving to the larger view. There are two visible strands that extend with decreasing zero-crossings and increasing coherency, which are indicative of distinct populations. The strands

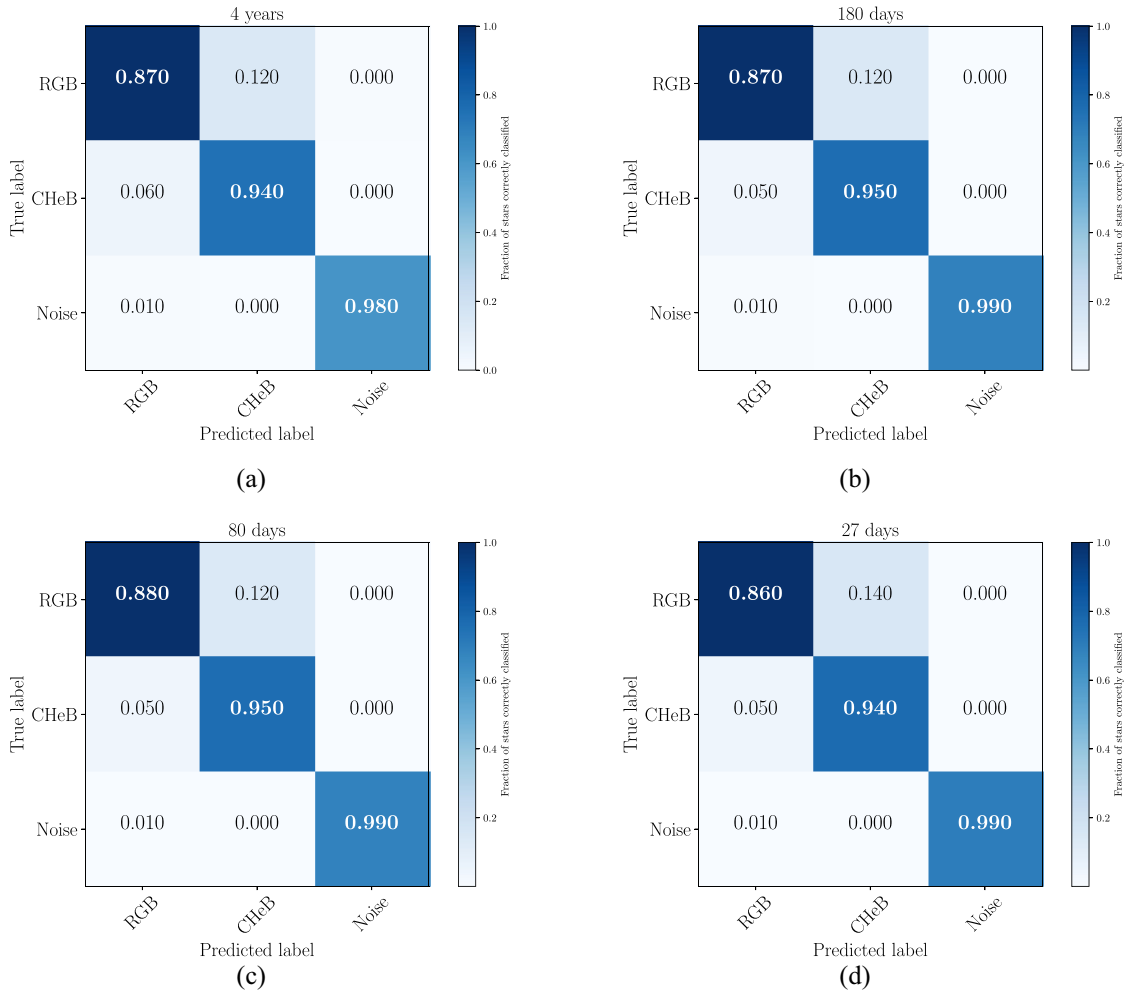

Figure 3. Confusion matrices for the classifier over each times-series length, starting with 4 yr in panel (a) to 27 d in panel (d).

cross at very long time-scales (i.e. a low number of normalized zero-crossings), with the lower strand crossing over the more coherent, higher strand at ∼0.1 in the normalized number of zero-crossings.

The visible separation leaves the impression that stars with different properties can lie on each of these strands and so we overplot the populations of known types to investigate where they fall in parameter space. This is shown in Fig. 5(b). We include stars showing rotation from McQuillan, Mazeh & Aigrain (2014), solar-like oscillators from Hon et al. (2019) that have a detection probability of 1, eclipsing binaries from the Kepler Eclipsing Binary Catalogue (Prša et al. 2011; Abdul-Masih et al. 2016; Kirk et al. 2016), and a sample of A and F-type stars that lie in the δ-Scuti instability strip that include δ-Scuti and γ-Doradus variables from Murphy et al. (2019).

The higher, more coherent strand is populated by stars exhibiting some degree of rotational modulation and the lower, less coherent strand is in fact populated by solar-like oscillators. This conforms with expectations, since solar-like oscillators are inherently stochastic and their oscillations are mostly much less coherent than signal from rotational modulation, and so for any given time-scale (inferred by proxy from the number of zero-crossings) a solar-like oscillator and rotational variable can be distinguished through the coherency of the signal. However, the strands do cross at a value of 0.1 in normalized zero-crossings. This is due to the fact that the granulation time-scale is closely linked to the radius of the star, the larger the radius of the star, the larger the convective cells which leads to a larger granulation time-scale, i.e. a decreasing number of zero-crossings. The rate of ascent of a star up the RGB is not linearly proportional to the number of zero-crossings. Therefore, we interpret this sudden rapid increase in the coherency of the signal as due to the increasingly fast evolution of the star towards the tip of the RGB.

Eclipsing binaries appear in Fig. 5(b) in two different configurations depending on the dominant contribution to the signal in the data. If the dominant contribution is stellar in origin, i.e. pulsations, then the star will appear on the strands following the stellar signal. However, if the dominant contribution is instead from the eclipses then they will lie away from the strands with a high coherency corresponding to the regularity of the eclipse signal and a normalized zero-crossing value proportional to the binary period.

Finally, we have the classical pulsators that appear at two different extremes of Fig. 5 due to the different pulsations observed. γ Dor pulsators are gravity-mode pulsators (where buoyancy is the restoring force of the pulsations) and so oscillate at long periods, this is evident by the low normalized number of zero-crossing values and high coherency of the signal. Whereas δ Sct pulsators are pressure-mode pulsators (where pressure is the restoring force of the pulsations) and oscillate at much higher frequencies placing them at the high extreme in normalized zero-crossings. Their signals in long cadence are seen

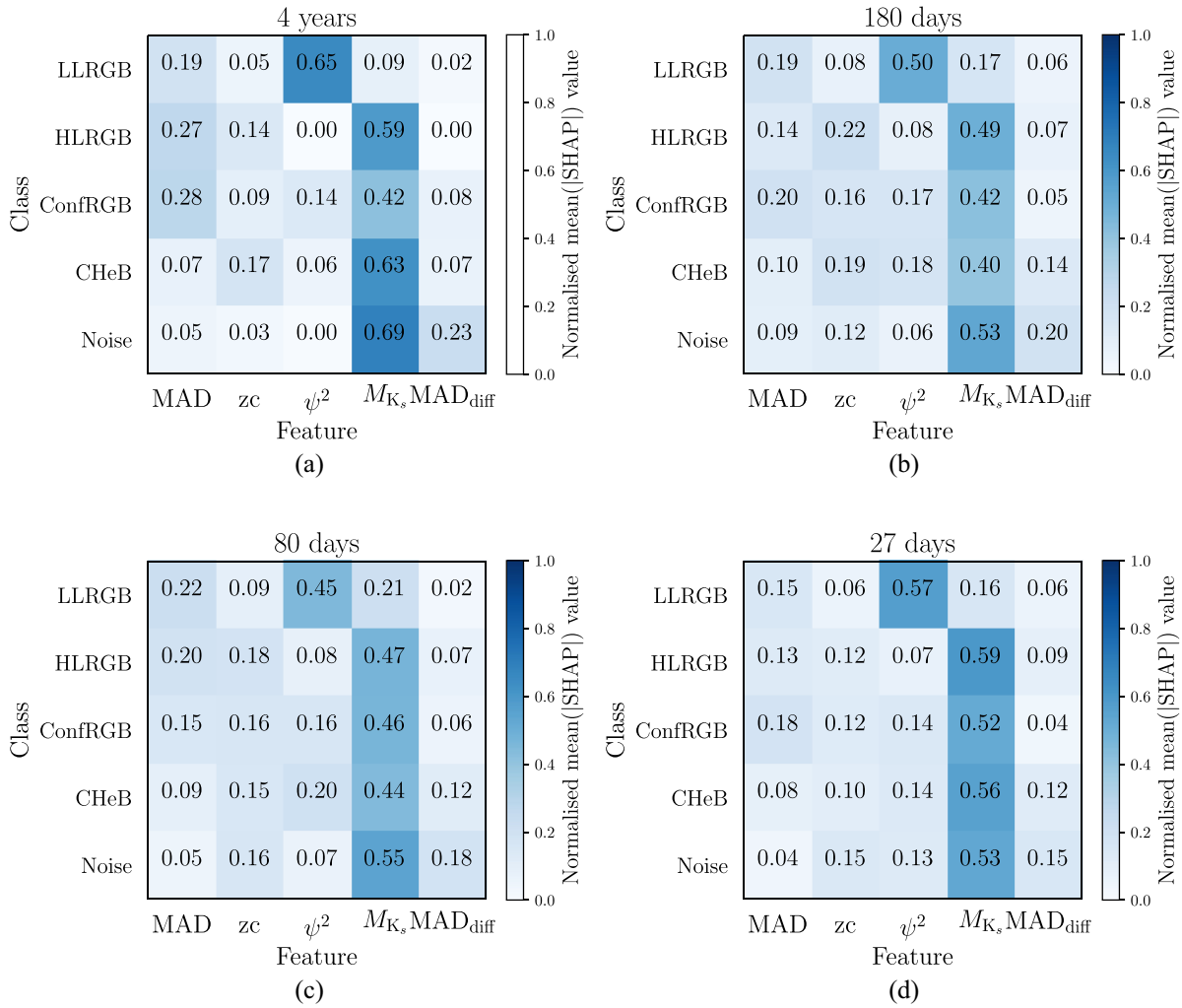

Figure 4. Feature matrices for each trained model, arranged according to descending time-series length. The normalized number of zero-crossings are labelled as 'zc'.

to be less coherent than the γ Dor pulsators, most likely due to the fact that they oscillate above the long-cadence Nyquist frequency (of $\sim$24 d^{-1}) and so it is likely that the lack of coherency comes from the undersampling of the data rather than from the oscillation modes themselves.

The investigation of the time-series features chosen in Section 2 applied to the full *Kepler* data set shows that not only can these features distinguish RGB from CHeB stars, but they could also potentially be used to classify a wider range of variable stars.

7 DISCUSSION

The trained classifiers perform very well over each data set length, however there are a few assumptions that we have made that should be addressed. Like any machine-learning task, the ability to generalize to unseen data can be affected by the quality of the training data. There could be an issue if the underlying populations between the new data and the training data differ significantly. We are therefore making the assumption that the underlying population observed by *Kepler* would be close to other observed underlying populations, for example, observed by K2 or *TESS*. Given that *Kepler* stared at a single patch of sky for 4 yr and other missions, such as K2 and *TESS*, look over more and larger patches of sky this assumption may not be strictly valid. This also applies to the properties of the particular mission. Time series that this classifier can be applied to must have a cadence that is similar to the training data. Otherwise this will introduce biases into the computed features and lead to incorrect classification. The feature that the differences between the training data and new unseen data have the largest effect on is the *K*-band absolute magnitude, whereby population level effects are ignored in its calculation (e.g. Girardi & Salaris 2001; Girardi 2016), using equation (12). Therefore, slight changes in the underlying M_{K_s} distribution could cause the predictions from the classifier to be affected. However, this is the reason why we use an intrinsic property rather than parallax and apparent magnitude, since absolute magnitude should be independent of distance. The other features could also be affected if the distribution of stellar parameters (e.g. mass, radius, metallicity, and effective temperature) is significantly different from those of the *Kepler* training set. This effect is easily quantifiable by comparing the training set feature distributions to those from a new data set. Secondly, we have not taken into account the difference in bandpasses between *Kepler* and *TESS* in which granulation and oscillation signals are expected to have lower amplitudes in *TESS* data due to the redder bandpass (Campante et al.

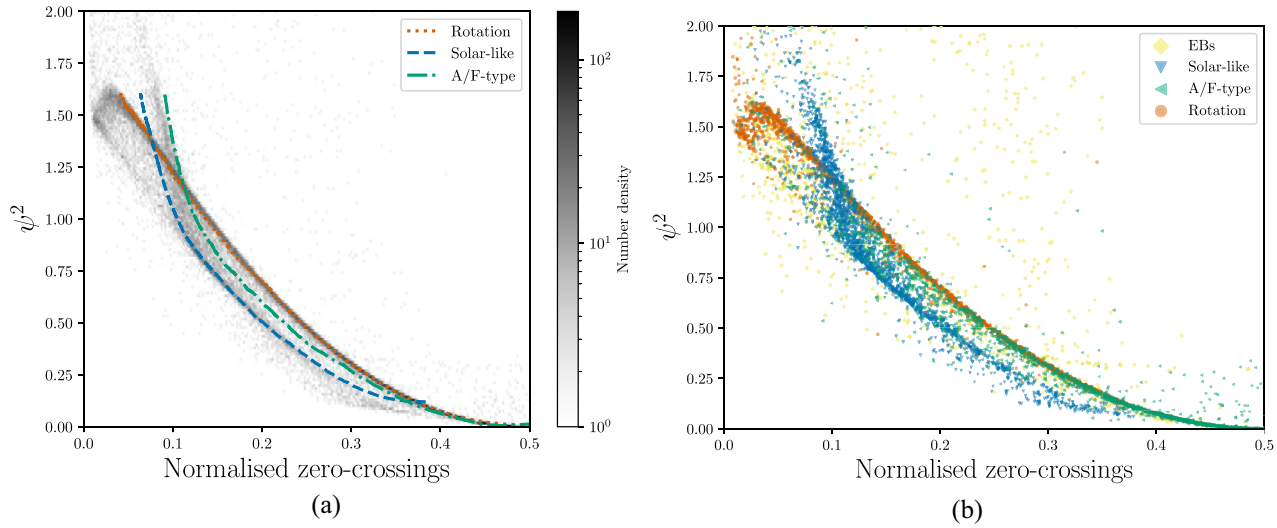

Figure 5. Panel (a) shows a hexbin plot showing the ψ^2 feature (coherency) as a function of the normalized zero-crossings for the full *Kepler* field, coloured by number density. Overplotted are the approximate positions of stars showing rotational modulation, solar-like oscillations, and δ-Scuti/γ-Doradus stars (labelled as A/F-type main-sequence stars) computed using a rolling mean of width 0.05 in normalized zero-crossings. Panel (b) shows the distribution of a random sample of 2000 stars in each category over the same region of parameter space as panel (a) with the addition of eclipsing binaries (EBs).

2016). Provided that the time series is dominated by granulation and physical signal then this can be approximately accounted for with a multiplicative factor in variance (Campante et al. 2016).

The fraction of RGB and CHeB stars in the training set compared to fraction of RGB and CHeB stars predicted are given in the two panels of Fig. 6. For the CHeB stars, we perform well and the distributions of stars in our training set and our predictions are very close. It can also be seen for the case of the secondary clump stars, the small hump around $\nu_{\rm max} \sim 80$ μHz, due to the agreement between the two distributions. Whereas for the RGB stars there is good agreement for the majority of $\nu_{\rm max}$ except for the region around the RC. There is a clear paucity of predicted RGB stars in the region $20 < \nu_{\rm max}(\mu{\rm Hz}) < 50$ which means that these RGB stars have been classified as CHeB stars. In addition, there is also a slight overabundance of RGB stars at $\nu_{\rm max} \sim 50$–60 μHz which means that CHeB stars have been misclassified as RGB stars. This is due to the class imbalance in the region around the RC, which can be as high as 7:1 in favour of CHeB stars at $\nu_{\rm max} \sim 30$ μHz.

There are two other cases that can result in a slightly poorer performance of the classifier which are not due to the classifier itself. The first is if the computed absolute magnitude is not correct, this could be due to source confusion during the cross-match to the *Gaia* data resulting in choosing the wrong distance estimate. In the event that the computed absolute magnitude is not correct, this will naturally cause the classifier to misclassify the star in question. This situation could arise with any of the other features, however it is most likely to occur for the absolute magnitude calculation due to source confusion. The second case is when the white noise level is very high, either due to a large amount of shot noise or because the star is very faint. As such the granulation signal can be very hard or impossible to detect which will result in the calculated features being in the wrong part of feature space. This will result in the star being classified as noise. It can to some degree be accounted for in the shorter length time series, whereby the signal-to-noise level is lower. However, more extreme situations cannot be accounted for in our training set.

Despite training our models on real data from *Kepler*, the described accuracy is likely an upper limit when applied to other data sets, e.g. K2, *TESS* or, in the future, *PLATO*. This is due to the different way in which pipelines detrend the data, which can affect the extraction of the zero-crossings (and HOCs).

In order for the probabilities returned by the classifier to be used in subsequent analyses, it is important for the probabilities to be calibrated. By calibrated we mean that if the probability returned for a star is 0.7, then we would expect this to be correct 70 per cent of the time. This means that the probability produced by the model is representative. It is well known that tree-based methods produced calibrated probabilities and so we do not include any post-processing to do so (see Appendix C).

8 CONCLUSION

In this work, we have trained a series of classifiers on differing length data sets that can robustly and accurately classify the evolutionary state of red giant stars. The classifiers make use of time-domain features: the MAD of the time series and its first differences, the normalized number of zero-crossings, and a coherency measure ψ^2. In addition to the time-domain features, the absolute *K*-band magnitude is also used. The trained models for each time-series length and the code used to extract the features are available at https://github.com/jsk389/Clumpiness. All classifiers, including that applied to the shortest data set length simulating the shortest data sets available from *TESS* achieve above 91 per cent accuracy when inferring red giant evolutionary states. As a result, our classifier will be highly applicable to classifying the large number of stars observed with *TESS* and, in the future, *PLATO* as well.

We have also shown that our probabilities are well calibrated and so can be readily applied to probabilistic analyses that require specific populations of stars, e.g. the RC, or that can be used as prior probabilities, e.g. in future peak-bagging codes (e.g. Corsaro 2019).

The time-series features that we propose are not only useful for determining evolutionary states, they are also of great use for

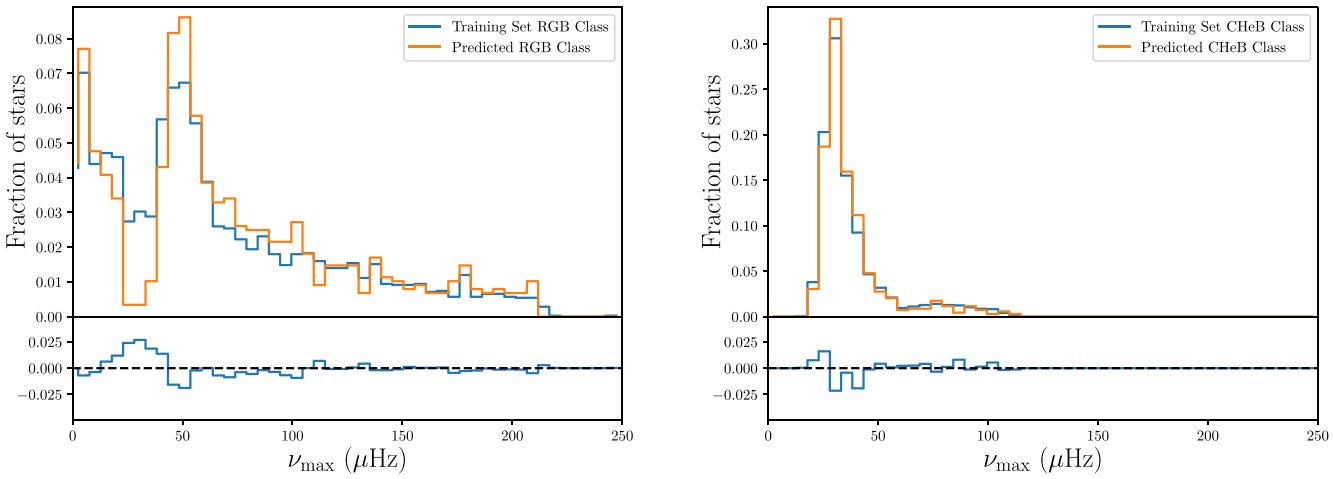

Figure 6. Comparison of the distribution of RGB stars (left) and CHeB stars (right) in the training set (blue) and the predictions (orange). The difference between the training set and prediction distributions is shown below each histogram.

identifying different stellar populations. This has been demonstrated briefly for a random subset of all the stars observed with *Kepler* and these can be of great use in subsequent classification tasks applied to multiple stellar populations. For example, both the normalized number of zero-crossings and the coherency measure ψ^2 are being used as features in the *TESS* Data for Asteroseismology classification work (Tkachenko et al., in preparation).

ACKNOWLEDGEMENTS

We would like to thank Marc Hon, Yvonne Elsworth, and Nathalie Themeßl for their very useful comments and discussions. The research leading to the presented results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no. 338251 (StellarAges). KJB is supported by the National Science Foundation under Award No. AST-1903828. This work has made use of data from the European Space Agency (ESA) mission *Gaia* (https://www.cosmos.esa.int/gaia), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, https://www.cosmos.esa.int/web/gaia/dpac/consortium). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement. This research has made use of the NASA Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

This work made use of the gaia-kepler.fun cross-match data base created by Megan Bedell.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Abdul-Masih M. et al., 2016, AJ, 151, 101
Akeson R. L. et al., 2013, PASP, 125, 989
Alcock C. et al., 1997, ApJ, 486, 697
Alcock C. et al., 2003, ApJ, 598, 597
Armstrong D. J. et al., 2015, A&A, 579, A19
Armstrong D. J. et al., 2016, MNRAS, 456, 2260
Bae J., Ryu Y., Chang T., Song I., Kim H. M., 1996, Signal Process., 52, 75
Baglin A., Auvergne M., Barge P., Deleuil M., Catala C., Michel E., Weiss W., COROT Team, 2006, in Fridlund M., Baglin A., Lochard J., Conroy L., eds, ESA SP-1306: The CoRoT Mission Pre-Launch Status – Stellar Seismology and Planet Finding. ESA, Noordwijk, p. 33
Bailer-Jones C. A. L., Rybizki J., Fouesneau M., Mantelet G., Andrae R., 2018, AJ, 156, 58
Bedding T. R. et al., 2011, Nature, 471, 608
Bergstra J., Yamins D., Cox D., 2013, in Dasgupta S., McAllester D., eds, Proceedings of Machine Learning Research Vol. 28, Proceedings of the 30th International Conference on Machine Learning. PMLR, Atlanta, Georgia, USA, p. 115
Bishop C. M., 2006, Pattern Recognition and Machine Learning (Information Science and Statistics. Springer-Verlag, Berlin, Heidelberg, Germany
Borucki W. J. et al., 2010, Science, 327, 977
Bovy J. et al., 2014, ApJ, 790, 127
Brown T. M., Gilliland R. L., Noyes R. W., Ramsey L. W., 1991, ApJ, 368, 599
Campante T. L. et al., 2016, ApJ, 830, 138
Chan V. C., Bovy J., 2020, MNRAS, 493, 4367
Chaplin W. J. et al., 2014, ApJS, 210, 1
Chen T., Guestrin C., 2016, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA
Christ M., Kempa-Liehr A. W., Feindt M., 2016, preprint (arXiv:1610.07717)
Corsaro E., 2019, Front. Astron. Space Sci., 6, 21
Cutri R. M. et al., 2003, 2MASS All Sky Catalog of Point Sources
Davies G. R. et al., 2017, A&A, 598, L4
Elsworth Y., Hekker S., Basu S., Davies G. R., 2017, MNRAS, 466, 3344
Elsworth Y. et al., 2019, MNRAS, 489, 4641
Friedman J., Tibshirani R., Hastie T., 2000, Ann. Stat., 28, 337
Gaia Collaboration, 2016, A&A, 595, A1
Gaia Collaboration, 2018, A&A, 616, A1
García R. A. et al., 2014, A&A, 568, A10
Girardi L., 2016, ARA&A, 54, 95
Girardi L., Salaris M., 2001, MNRAS, 323, 109
Girardi L., Groenewegen M. A. T., Weiss A., Salaris M., 1998, MNRAS, 301, 149

Green G. M. et al., 2015, ApJ, 810, 25
Groenewegen M. A. T., 2008, A&A, 488, 935
Guo C., Pleiss G., Sun Y., Weinberger K. Q., 2017, Proceedings of the 34th International Conference on Machine Learning - Volume 70. JMLR.org, Sydney, NSW, Australia, p. 1321
Hall O. J. et al., 2019, MNRAS, 486, 3569
Hastie T., Tibshirani R., Friedman J., 2001, The Elements of Statistical Learning. Springer, New York, NY, USA
Hawkins K., Leistedt B., Bovy J., Hogg D. W., 2017, MNRAS, 471, 722
Hekker S., Barban C., Baudin F., De Ridder J., Kallinger T., Morel T., Chaplin W. J., Elsworth Y., 2010, A&A, 520, A60
Hekker S. et al., 2012, A&A, 544, A90
Hekker S., Elsworth Y., Basu S., Bellinger E., 2017, EPJ Web Conf., 160, 04006
Hon M., Stello D., Yu J., 2018, MNRAS, 476, 3233
Hon M., Stello D., García R. A., Mathur S., Sharma S., Colman I. L., Bugnet L., 2019, MNRAS, 485, 5616
Howell S. B. et al., 2014, PASP, 126, 398
Huber D. et al., 2017, ApJ, 844, 102
Jenkins J. M. et al., 2010, ApJ, 713, L120
Kallinger T. et al., 2012, A&A, 541, A51
Kedem B., Slud E., 1981, Biometrika, 68, 551
Kedem B., Slud E., 1982, Ann. Stat., 10, 786
Khan S. et al., 2019, A&A, 628, A35
Kim D.-W., Bailer-Jones C. A. L., 2015, Astrophysics Source Code Library, record ascl:1512.019
Kirk B. et al., 2016, AJ, 151, 68
Kjeldsen H., Bedding T. R., 1995, A&A, 293, 87
Kügler S. D., Gianniotis N., Polsterer K. L., 2015, MNRAS, 451, 3385
Leys C., Ley C., Klein O., Bernard P., Licata L., 2013, J. Exp. Soc. Psychol., 49, 764
Lightkurve Collaboration, 2018, Astrophysics Source Code Library, record ascl:1812.013
Lindegren L. et al., 2018, A&A, 616, A2
Lundberg S. M., Lee S.-I., 2017, in Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., eds, Advances in Neural Information Processing Systems 30. Curran Associates, Inc., Red Hook, NY, USA, p. 4765
Lundberg S. M., Erion G. G., Lee S.-I., 2018, preprint (arXiv:1802.03888)
McQuillan A., Mazeh T., Aigrain S., 2014, ApJS, 211, 24
Mosser B. et al., 2011, A&A, 532, A86
Mosser B. et al., 2014, A&A, 572, L5
Murphy S. J., Hey D., Van Reeth T., Bedding T. R., 2019, MNRAS, 485, 2380
Naul B., Bloom J. S., Pérez F., van der Walt S., 2018, Nat. Astron., 2, 151
Nun I., Protopapas P., Sim B., Zhu M., Dave R., Castro N., Pichara K., 2015, preprint (arXiv:1506.00010)
Paczyński B., Stanek K. Z., 1998, ApJ, 494, L219
Pinsonneault M. H. et al., 2018, ApJS, 239, 32
Pojmanski G., 2002, Acta Astron., 52, 397
Prša A. et al., 2011, AJ, 141, 83
Rauer H. et al., 2014, Exp. Astron., 38, 249
Ricker G. R. et al., 2015, J. Astron. Telesc. Instrum. Syst., 1, 014003
Skrutskie M. F. et al., 2006, AJ, 131, 1163
Stello D. et al., 2013, ApJ, 765, L41
Udalski A., Olech A., Szymanski M., Kaluzny J., Kubiak M., Krzeminski W., Mateo M., Stanek K. Z., 1996, Acta Astron., 46, 51
Udalski A. et al., 2008, Acta Astron., 58, 329
Valenzuela L., Pichara K., 2018, MNRAS, 474, 3259
Vrard M., Mosser B., Samadi R., 2016, A&A, 588, A87
Youden W. J., 1950, Cancer, 3, 32
Yu J., Huber D., Bedding T. R., Stello D., Murphy S. J., Xiang M., Bi S., Li T., 2016, MNRAS, 463, 1297
Zinn J. C., Pinsonneault M. H., Huber D., Stello D., 2019, ApJ, 878, 136

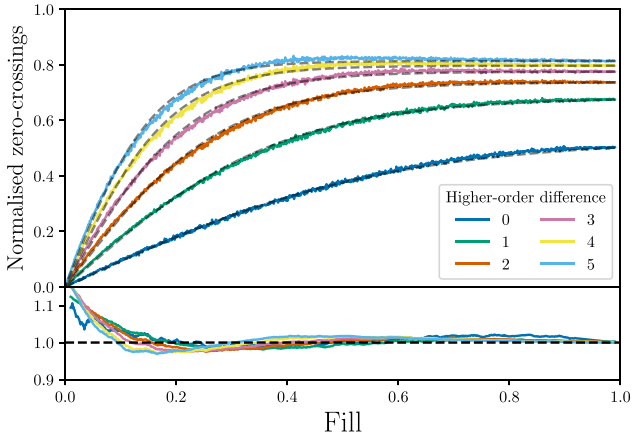

Figure A1. The number of zero-crossings in a white noise time series normalized by the number of data points in the data, plotted as a function of fill. Each line is coloured according to the higher order difference calculated, where 0 is the equivalent to the zero-crossings of the original time-series data. The normalization functions for each higher order difference are given by the grey dashed lines. The bottom panel shows the residuals of the normalization function fits to the data in the top panel, smoothed over 20 points for clarity and coloured by the corresponding higher order difference.

APPENDIX A: ACCOUNTING FOR FILL IN THE COMPUTATION OF HIGHER ORDER CROSSINGS

The presence of gaps in the time series will affect the estimation of the number of zero-crossings, and subsequent higher order differences, causing an underestimation of the underlying value of the number of zero-crossings. A correction therefore needs to be applied to account for gaps in the time-series data so that we can extract reliable zero-crossing estimates from data sets with missing data. The purpose of such a correction should be that when applied, the number of zero-crossings estimated accounting for the fill accurately reflects the underlying number of zero-crossings if the data had no gaps.

The correction factor was determined using a number of simulated white noise time series. Fig. A1 shows the results of the simulations for a variety of fill values and higher order differences. In order to produce data with differing numbers of gaps, data points were randomly selected and set to zero, in accordance with the desired fill value. As a result only short time-scale gaps were inserted.

The shape of the curves in the top panel of Fig. A1 resembles half of a sigmoid function, which provides a straightforward analytical formulation that can be fitted to the data. We define the normalization function as follows:

$$S(x) = \frac{1}{1 + \exp(-kx)} - 0.5, \quad \text{for } x > 0, \tag{A1}$$

where x is the value of the fill and k is a free parameter that is fitted to the data.

The subtraction of 0.5 ensures $S(x = 0) = 0$, which is a condition set by our data since when there are no data points (i.e. all zeros), there are no zero-crossings. However in its current guise, the above function is not quite suitable because it varies between 0 and 0.5 for positive x. In order to extend it to the full range over the number of normalized zero-crossings, we normalize it by $S(x = 1)$, the value of the function for full fill, and then multiply it by the number of

Table B1. The `xgboost` hyperparameter values and hyperparameter space chosen for tuning.

	Length of data set				Tuning	
Hyperparameter	4 yr	180 d	80 d	27 d	Range	Step
η	0.1	0.375	0.275	0.025	0.025–0.5	0.025
γ	7.5	7.5	7.5	7.5	–	–
`max_depth`	3	6	10	8	1–13	1
`min_child_weight`	4	1	2	3	1–6	1
`subsample`	0.7	0.5	0.85	0.5	0.5–1.0	0.05
`colsample`	0.7	0.5	0.85	0.85	0.5–1.0	0.05

normalized zero-crossings at full fill for each HOC. Examples of this function fitted to the data are shown in Fig. A1, which was performed by minimizing the residual sum of squares.

The fill correction is a multiplicative factor and so to discern the accuracy of the fitted values, we look at the ratio of the data to the fill correction as a function of fill, as shown in the bottom panel of Fig. A1. Note the scale of the 'residuals' which in turn shows that we are precise to ∼5 per cent (for the worst case) at a fill of 0.4. Due to the additional data pre-processing and the merging of data with large gaps in between, the fill of the real data rarely falls below ∼0.8.

APPENDIX B: HYPERPARAMETER TUNING

There are a number of hyperparameters of the model that require tuning in order to achieve a high degree of accuracy.

(i) the learning rate $\eta \in [0, 1]$ dictates how much the contribution of the newest tree is scaled relative to the current model, i.e. how much of the variance of residuals it is fitted to. A lower value removes less variance from the residuals and indicates a more conservative model.

(ii) `max_depth` controls the maximum depth of each tree, i.e. how many times an individual tree undergoes a split.

(iii) `min_child_weight` controls the conditions under which a split occurs. Only if the sum of the weights at a given split exceeds this value does the tree continue to grow.

(iv) `subsample` dictates the fraction of training data used to grow each tree.

(v) `colsample` that dictates the fraction of features used when constructing each tree. This means that not all features are necessarily used when constructing each tree.

(vi) λ is the L2 regularization term on the model weights and is fixed to the default value of 1.

(vii) α is the L1 regularization term on the model weights and is fixed to the default value of 0.

The degree of regularization, γ, which penalizes the complexity of each constructed tree is not included in the hyperparameter tuning. This is because a value of zero, i.e. no regularization, will always be chosen as introducing regularization will inherently increase the minimum loss value that can be obtained. The value of γ was chosen such that the difference between the accuracy of the training and validation sets were minimized, i.e. the model generalizes well to the unseen validation data. The value of γ was fixed across all lengths of data set to a value of 7.5.

We use the HYPEROPT package (Bergstra et al. 2013) to perform the search over the hyperparameter space defined in Table B1. The HYPEROPT package then finds the combination of hyperparameters that minimizes a chosen objective function using a random search. This function is chosen to be the multiclass log-loss (as given in equation 15) and 250 trials are made to find the best combination of hyperparameters. The search is performed separately for each time-series length.

APPENDIX C: PROBABILITY CALIBRATION

As stated previously, the classifier returns the probability that a star belongs to a given class rather than just the class label. This gives the possibility for the probability produced by the classifier to be used in subsequent probabilistic analyses, whether that be as a prior distribution on RC membership or perhaps in future peak-bagging codes. In order for these probabilities to be used, they need to be well-calibrated so that they can be used as prior or posterior probabilities.

The classifier produces a probability that a star belongs to a given class that we shall call the 'confidence', if the classifier assigns a confidence of 0.5–100 predictions then we would expect 50 of the predictions to be correct. If this is the case then the probabilities are said to be 'calibrated'.

It is important to assess whether the probabilities outputted by the classifier are well-calibrated. In this work, we use temperature scaling applied to the multiclass classification problem from Guo et al. (2017)[4] to test the calibration of the probabilities. The way in which the classifier obtains class probabilities is by predicting the logits (the logarithm of the odds that a star belongs to a particular class) and passing them through a softmax function

$$\sigma_{\rm SM}(\mathbf{z}_i)^{(k)} = \frac{\exp\left(z_i^{(k)}\right)}{\sum_{j=1}^{K}\exp\left(z_i^{(j)}\right)}, \tag{C1}$$

where K is the number of classes in the classification problem, the index i represents an individual star, and $\mathbf{z}_i$ is the logits for a given star. The class prediction probability is then made by

$$\hat{p}_i = \max_k \sigma_{\rm SM}(\mathbf{z}_i)^{(k)}. \tag{C2}$$

The aim of temperature scaling is to essentially 'soften' the softmax function changing the output probabilities, but not changing to the overall accuracy of the model. In other words, we want to obtain new calibrated probabilities $\hat{q}$ such that

$$\hat{q}_i = \max_k \sigma_{\rm SM}(\mathbf{z}_i/T)^{(k)}, \tag{C3}$$

where T is the temperature that is a fitted parameter post-classifier training (see Guo et al. 2017 for more details).

[4] https://github.com/gpleiss/temperature_scaling

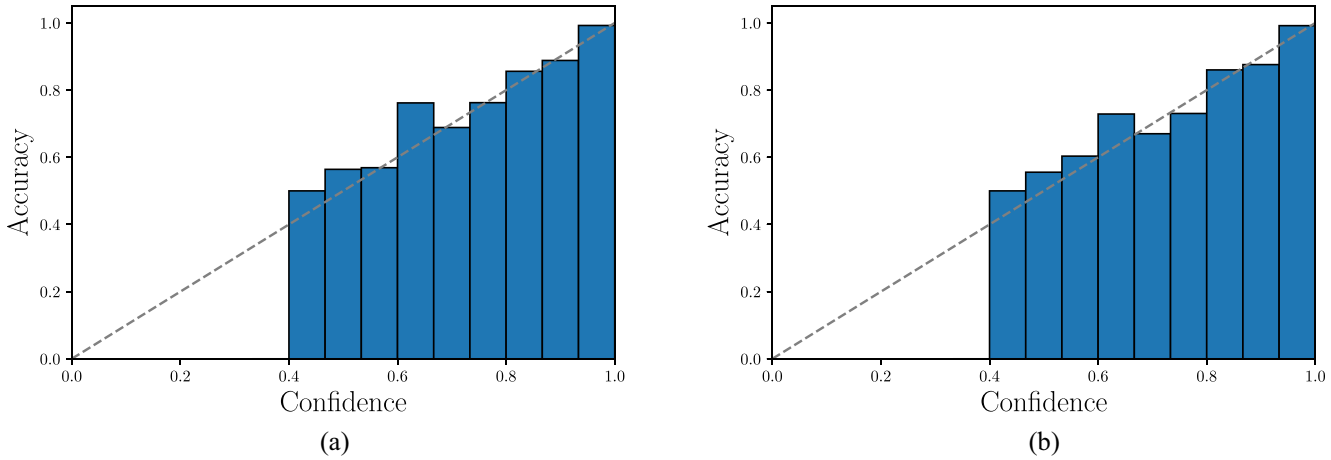

Figure C1. A reliability plot for the 27 d classifier shown before the probability calibration (a) and after the calibration (b). The 1:1 diagonal is given by the grey dashed line.

The calibration can be summarized in a reliability plot, as shown in Fig. C1, which displays the binned confidence of the predictions from a classifier as a function of the accuracy of the classifier in that bin. If the probabilities are perfectly calibrated then the data will lie along the 1:1 diagonal, and any deviation will show some degree of miscalibration. Fig. C1(a) shows the reliability plot of the 27 d classifier before the probability calibration and it can be seen that the probabilities are well calibrated. Once the probability calibration has been applied the difference is only very slight. As a result we choose not to add this extra post-processing step to the classifier probabilities and confirm that the probabilities can be reliably interpreted in subsequent analyses.

This paper has been typeset from a TEX/LATEX file prepared by the author.