# Vetting Asteroseismic $\Delta\nu$ Measurements using Neural Networks

Claudia Reyes[1]★, Dennis Stello[1,2,3], Marc Hon[4,1] and Joel C. Zinn[5,1]†
[1] School of Physics, University of New South Wales, NSW 2052, Australia
[2] Sydney Institute for Astronomy (SIfA), School of Physics, University of Sydney, NSW 2006, Australia
[3] Stellar Astrophysics Centre, Department of Physics and Astronomy, Aarhus University, DK-8000 Aarhus C, Denmark
[4] Institute for Astronomy, University of Hawaii, 2680 Woodlawn Drive, Honolulu, HI 96822, USA
[5] Department of Astrophysics, American Museum of Natural History, Central Park West at 79th Street, New York, NY 10024, USA

Accepted XXX. Received YYY; in original form ZZZ

## ABSTRACT

Precise asteroseismic parameters allow one to quickly estimate radius and mass distributions for large samples of stars. A number of automated methods are available to calculate the frequency of maximum acoustic power ($\nu_{\rm max}$) and the frequency separation between overtone modes ($\Delta\nu$) from the power spectra of red giants. However, filtering through the results requires either manual vetting, elaborate averaging across multiple methods, or sharp cuts in certain parameters to ensure robust samples of stars free of outliers. Given the importance of ensemble studies for Galactic archaeology and the surge in data availability, faster methods for obtaining reliable asteroseismic parameters are desirable. We present a neural network classifier that vets $\Delta\nu$ by combining multiple features from the visual $\Delta\nu$ vetting process. Our classifier is able to analyse large numbers of stars determining whether their measured $\Delta\nu$ are reliable thus delivering clean samples of oscillating stars with minimal effort. Our classifier is independent of the method used to obtain $\nu_{\rm max}$ and $\Delta\nu$, and therefore can be applied as a final step to any such method. Tests of our classifier's performance on manually vetted $\Delta\nu$ measurements reach an accuracy of 95%. We apply the method to giants observed by K2 Galactic Archaeology Program and find that our results retain stars with astrophysical oscillation parameters consistent with the parameter distributions already defined by well-characterised *Kepler* red giants.

**Key words:** asteroseismology – stars: oscillations – stars: fundamental parameters – methods: data analysis

## 1 INTRODUCTION

Since the launch of CoRoT (Baglin et al. 2000) and *Kepler* (Borucki et al. 2010; Koch et al. 2010), asteroseismic analysis pipelines such as SYD (Huber et al. 2009), COR (Mosser & Appourchaux 2009), CAN (Kallinger et al. 2010), A2Z (Mathur et al. 2010; García et al. 2014), BAM (Zinn et al. 2019), and BHM (Elsworth et al. 2017) have been developed to extract $\nu_{\rm max}$ and $\Delta\nu$ in more automated ways than was done in the past. To analyse the data and to determine $\Delta\nu$, each of these pipelines relies on different methods such as: the autocorrelation of the power spectrum (SYD), the autocorrelation of the timeseries (COR) or equivalently, the power spectrum of the power spectrum (BHM), a fit to the power spectrum (CAN), a fit to the folded power spectrum (BAM), or a combination thereof (A2Z, BAM). Additional statistical testing is also used by some as internal detection calibration. However, many of the pipelines still require a form of vetting to remove unreliable measurements of $\Delta\nu$ beyond what is captured by statistical significance testing. The vetting therefore often involves some sort of manual verification sometimes involving results from multiple pipelines and/or hard-coded cuts in certain parameters. The former is very time consuming and the latter can easily result in unphysical sharp features in the properties of the resulting stellar population, which can be undesirable.

After *Kepler* came K2 (Howell et al. 2014), and for the first time a constant flow of large amounts of data of previously unknown seismic targets that needed vetting beyond what is suitably performed using a manual approach. With the launch of TESS (Ricker et al. 2014) and later in this decade of PLATO (Rauer 2017), fully automated yet robust methods are more necessary than ever to ensure fast and reliable asteroseismic measurements providing both complete and pure sets of measurements.

Here we present a neural network-based classifier that is able to determine whether $\Delta\nu$ values are reliable, independent of the method used to derive $\Delta\nu$ and $\nu_{\rm max}$. We start by giving an overview of the data used in this paper. Then we describe the methods used to build the machine learning model, and next we show its performance on the training set using traditional machine learning performance metrics. Finally, we examine the classifier's performance on data from different pipelines by comparing our vetted results with $\nu_{\rm max}$ and $\Delta\nu$ distributions reported by Zinn et al. (2021).

## 2 DATA

The data used in this project correspond to observations obtained as part of the K2 Galactic Archaeology Program, GAP, Campaigns 1 to 8 and 10 to 18 (Stello et al. 2017; Zinn et al. 2020; Zinn et al. 2021). K2 GAP targets were chosen to satisfy simple colour and magnitude cuts where red giants are more likely to be found (Stello et al. 2015;

★ E-mail: claudiarreyes@icloud.com
† NSF Astronomy and Astrophysics Postdoctoral Fellow

Sharma et al. 2021). Reasons for targeting red giants are: their high luminosity, which allows probing deeper Galactic regions, and their oscillation frequencies, which are detectable from K2 long-cadence data (which has a Nyquist frequency ≈ 280 $\mu$Hz). Hence solar like oscillators from K2 GAP are expected to be mostly red giant branch (RGB) stars in various evolutionary phases as well as core Helium burning red clump (RC) stars.

For the initial analysis of this paper, values of $\Delta\nu$ and $\nu_{\rm max}$ are from the SYD pipeline, while in Sections 4.2 we examine the results of our method on results from various other pipelines in literature.

## 3 METHODOLOGY

We want to build a classifier that can vet $\Delta\nu$ regardless of the pipeline that provided the measurement. For this task we choose to use supervised learning, a machine learning technique characterised by its use of labelled data sets to train algorithms for data classification (or regression). Specifically, we use neural networks because we need a method that can deal with the various aspects and complexities shown by oscillation spectra of red giants. Key capabilities of such non-linear algorithms, including parallel processing of multiple features and complex-pattern detection and extrapolation, make neural networks well-suited for this task.

We first discuss the data set used to train our machine learning method in Section 3.1, then we discuss the features we extract from power spectra in Section 3.2, the neural network architecture in Section 3.3, the training steps in Section 3.4 and finally the assessment of the network's performance on the training data in Section 3.5.

### 3.1 Training Set Preparation

It is essential to carefully build a training set that is balanced and representative of the different spectra that our machine learning algorithm will encounter in practice. Following Yu et al. (2018) we use visual inspection to classify stars as having $\Delta\nu$ detections or not. To prepare for the visual vetting we generate diagnostic plots that allow us to examine the identified oscillations in power spectra using the following three diagrams.

#### 3.1.1 *Autocorrelation function*

The first diagnostic we use is the autocorrelation function (Figure1b) of the power spectrum (Figure1a). The autocorrelation highlights the near regularity of the oscillation spectrum and is useful to confirm if $\Delta\nu$ can be measured reliably and whether it has been measured correctly. This regularity is expected from the asymptotic relation (Tassoul 1980), where the frequency of a mode with spherical degree $l$ of radial order $n$ is given by

$$\nu_{n,l} \approx \Delta\nu(n + \frac{l}{2} + \epsilon) - \delta\nu_{0,l}. \quad (1)$$

Here $\epsilon$ is a dimensionless offset or phase term and the small separation $\delta\nu_{0,l}$ is defined as 0 for $l = 0$. From equation 1 we expect the autocorrelation to show peaks at multiples of $\sim\Delta\nu/2$.

To produce the plots we calculate the autocorrelation up to a shift of the spectrum equivalent to three $\Delta\nu$ and then we scale the amplitudes between 0 and 1, but first we make sure to avoid the global maximum of the autocorrelation (at a shift of zero) by ignoring the first 0.02$\Delta\nu$ shift. To do this, we standardise the function to a fixed length ensuring that the autocorrelation peaks were not smoothed away. A length of 300 data points was found to be conservative, therefore the first 6 points are ignored for each star.

**Table 1.** Nominal $\nu_{\rm max}$ values and calculated $\Delta\nu$ for the nine theoretical oscillation models considered to span the entire frequency range of the K2 sample. The last column shows the range of the observed $\Delta\nu$ we used to select which model corresponds to each star.

| Model | $\nu_{\rm max}$ [$\mu$Hz] | $\Delta\nu_{\rm model}$ [$\mu$Hz] | $\Delta\nu_{\rm obs}$ range [$\mu$Hz] |
|---|---|---|---|
| A | 350.5 | 24.5 | [ 18.9, 31.5 ) |
| B | 180.8 | 14.7 | [ 11.4, 18.9 ) |
| C | 94.6 | 8.9 | [ 7.65, 11.4 ) |
| D | 64.2 | 6.6 | [ 5.35, 7.65 ) |
| E | 37.4 | 4.4 | [ 3.44, 5.35 ) |
| F | 20.2 | 2.7 | [ 2.08, 3.44 ) |
| G | 10.0 | 1.6 | [ 1.25, 2.08 ) |
| H | 5.5 | 1.0 | [ 0.88, 1.25 ) |
| I | 4.0 | 0.8 | [ 0.71, 0.88 ) |

#### 3.1.2 *Folded Spectrum*

The second diagnostic is the folded oscillation spectrum (Figure1c). It is constructed by taking the central portion of the spectrum around the frequency of maximum power and co-adding each $\Delta\nu$ segment. This provides a simple way of showing the regularity in the mode pattern, and is particularly useful for low S/N cases where not every segment on their own would necessarily show the full pattern of modes.

To further guide the eye towards the expected pattern, we use model spectra that we fold and lay on top of the observed folded spectra (see grey dotted lines in Figure1c). If a reliable measurement of $\Delta\nu$ is used, the overall shape described by the folded spectrum will follow this template. We used theoretical oscillation modes for 1M$_\odot$ models in different evolutionary phases from the base to the tip of the RGB taken from Stello et al. (2014). These models were based on simulations from the ASTEC stellar evolution code (Christensen-Dalsgaard 2008), which does not include the later core Helium burning phases.

Table 1 lists the parameters of the models, from model A ($\nu_{\rm max}$=260.5$\mu$Hz) to model I ($\nu_{\rm max}$=4.0$\mu$Hz) used to cover the entire range of frequencies in the K2 sample. We derived $\Delta\nu$ from the radial modes following the approach by White et al. (2011), performing a weighted linear fit to the radial frequencies $l = 0$ as a function of the order $n$ as in equation 1, with weights obtained from a Gaussian window of width=0.25$\nu_{\rm max}$ centred on $\nu_{\rm max}$, taking the slope of this fit as our $\Delta\nu$. The last column of Table 1 refers to the range of $\Delta\nu$ from real stars for which we use each model. The boundaries for each range are set to the midpoint between $\Delta\nu$ of the models on a logarithmic scale. Note that these models fully take into account the presence of mixed modes that arise from the coupling between pressure and gravity waves, as it is evident specially for $l = 1$ from Figure 2a, in blue. The peak height of each mode is modelled as the inverse of the square root of the mode's inertia and scaled to that of the radial modes interpolated at each mode frequency, following Aerts et al. (2010) and Stello et al. (2014).

Figure 2a shows the central four $\Delta\nu$ segments around $\nu_{\rm max}$ of the modelled oscillation spectrum for models B, D, E and F, which represent four of the most commonly used models. Because for very low $\nu_{\rm max}$ models there are only significant oscillation modes in the four central $\Delta\nu$ segments around $\nu_{\rm max}$ (Stello et al. 2014, Figure 1), we obtain the folded spectra in Figure 2b from these four segments. For the real data, however, we consider six central segments to account

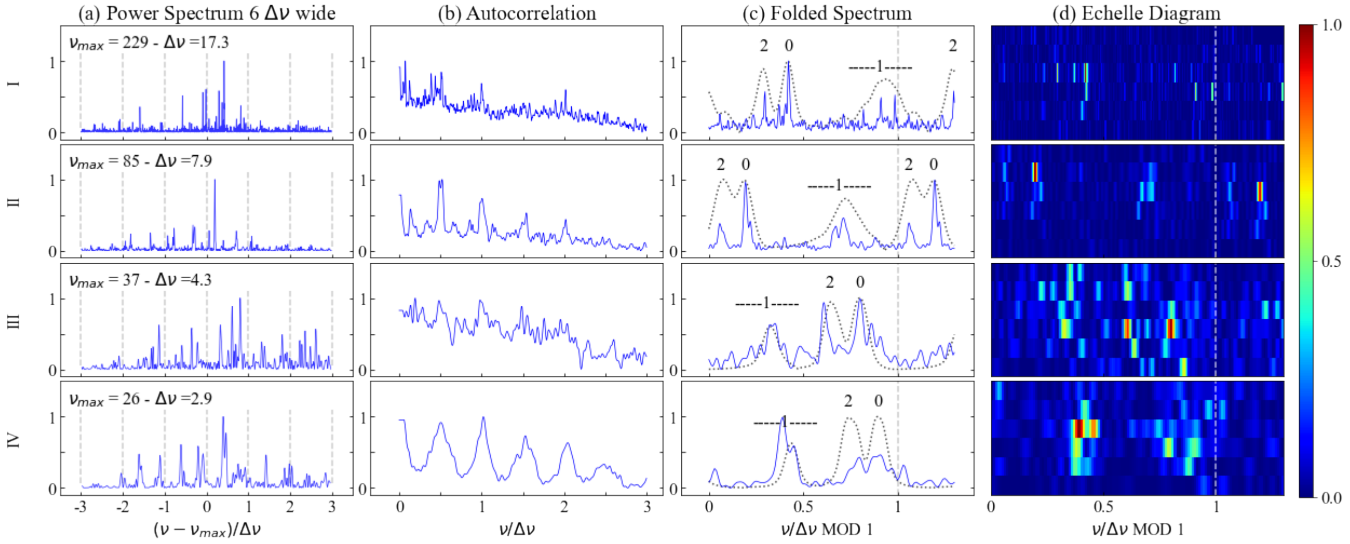

**Figure 1.** Examples of diagnostic plots from our training set showing stars with reliable Δ$\nu$ in different $\nu_{\rm max}$ ranges. Rows marked I-IV correspond to stars from K2 GAP campaign 1: EPIC 201829369, 201207669, 201245474, 201681005 respectively. (a) portion of the power spectrum around the frequency of maximum power $\nu_{\rm max}$. Segmented lines indicate multiples of Δ$\nu$. Here $\nu_{\rm max}$ has been shifted to the nearest multiple of Δ$\nu$. Annotated $\nu_{\rm max}$ and Δ$\nu$ values are given in $\mu$Hz.(b) autocorrelation function showing the periodic peaks at multiples of Δ$\nu$, (c) folded spectrum of width 1Δ$\nu$ obtained from folding the six central Δ$\nu$ segments from (a), shown here together with the folded modelled spectrum (grey dotted outline) assigned to the star based on its Δ$\nu$. Modes $l = 0, 1, 2$ are annotated. Modes of degree $l = 1$ appear spread out showing the coupling of dipoles with a large number of higher order g modes from the core. (d) The échelle diagram constructed from the power spectrum from column (a), with hotter colours indicating higher relative power. The extension to the right of the grey line marking 1$\nu_{\rm max}$ on (c) and (d) mirrors the first 30% of the diagram, for continuity. All data are scaled in amplitude between 0 and 1.

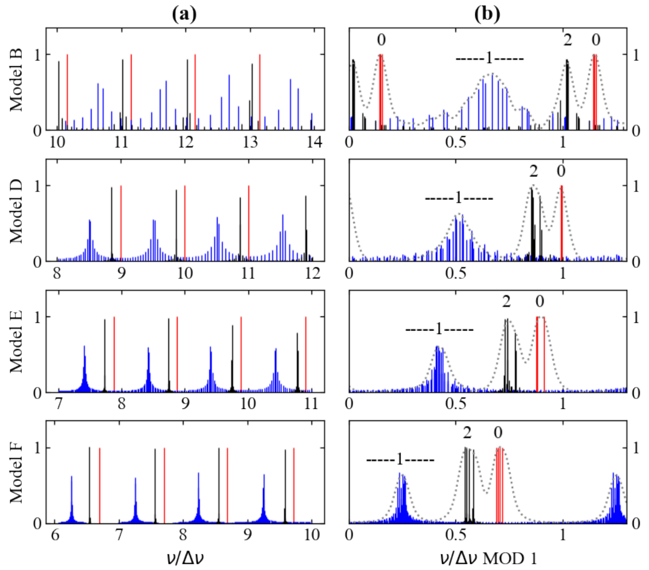

**Figure 2.** Models B, D, E and F from Table 1 representing four of the most commonly used models. Radial modes are shown in red, dipole modes in blue and quadrupole modes in black. All amplitudes are normalised. (a) four central segments of the spectra of width 1Δ$\nu$ around the frequency of maximum oscillation power $\nu_{\rm max}$. (b) folded model spectra obtained from folding (a). The positions of the $l = 0, 1, 2$ modes are annotated. The ordinate range of the folded spectrum is extended beyond 1 for clarity of the repeating structure of the oscillations.

for possible asymmetric distributions of the oscillation power around the value of $\nu_{\rm max}$.

To obtain the templates corresponding to the dotted lines in Figure 2b, we generate an array (not shown) in which we represent each of the modes with a Lorentzian of width 0.04Δ$\nu$. Finally, we convolve the upper envelope of this array with a Gaussian of $\sigma = 7\ \nu/\Delta\nu$ modulo 1, chosen to produce a smoothed template without losing any of the general features of the folded models. The resulting template is scaled to a fixed range between 0 and 1 and used together with the star's folded spectrum as in Figure 1c, where the template has been shifted to the position of maximum correlation. This shift takes care of the difference in $\epsilon$ between the data and the model (White et al. 2011). The choice of which template model to use is made according to each star's Δ$\nu$ and Table 1. Because solar-like oscillations are stochastically driven, the oscillation amplitudes seen in short time series (such as the 80-day K2 data used in this work) can show significant variation from the simple, and rather regular, mode inertia-based 'amplitudes' we use in the template. Hence we allow for some variation of the mode heights around the predicted model when we decide whether Δ$\nu$ is reliable.

### *3.1.3 Échelle diagram*

The third diagnostic plot is the échelle diagram, which is created by dividing the power spectrum into segments of length Δ$\nu$ and stacking each segment above one another. The resulting two-dimensional array is colour-coded according to power in each array bin. If a reliable measurement of Δ$\nu$ is used, the $l = 0$ and $l = 2$ modes are expected to align vertically in the échelle diagram (Bedding 2011). Figure 1d shows the diagram created from the central 6 Δ$\nu$ segments of the power spectrum. The échelle diagram carries additional information that the autocorrelation and folded spectra do not provide; in particular, while the natural curvature of the mode pattern can slightly blur the autocorrelation function and the folded spectrum, it can be displayed neatly in the échelle diagram (Kallinger et al. 2012, Figure 6). Still, the three diagrams in concert was useful in the process of visual vetting, particularly for spectra with missing modes, low S/N or full of mixed modes where one diagram alone might be inconclusive.

### 3.1.4 Constructing the training set

Having all three diagnostic plots in place we then start creating the training set. To obtain a representative sample we use stars from a wide range of K2 campaigns. We chose the 15,585 stars observed during campaigns 1, 4, 8, 13 and 15, that were deemed to be potentially oscillating giants by the machine learning algorithm from Hon et al. (2018b). In order to obtain a series of $\Delta\nu$ values that we could label as reliable or unreliable we ran the SYD pipeline on this full sample of 15,585 spectra and removed only those with values in the following ranges: $\Delta\nu \leq 0.3$ $\mu$Hz, $\nu_{\rm max} \leq 3$ $\mu$Hz, and $\nu_{\rm max} \geq 278.8$ $\mu$Hz, but retained all 15,170 remaining stars irrespective of the data actually showing oscillations or not. This was to have a significant fraction of $\Delta\nu$ values that we would later label visually as unreliable with the aim of having a training set with roughly equal numbers of reliable and unreliable labeled $\Delta\nu$ values.We note that the source of the $\Delta\nu$ values is not important for our training and subsequent results, as long as the final number of reliable and unreliable $\Delta\nu$ values ends up being balanced. In order words, we could have used mock-generated $\Delta\nu$ values.

For all the stars in the selected sample, we generated the three diagnostic plots, performed visual checks individually for each star, and from this concluded that a total of 7,240 stars showed oscillation mode structure consistent with a correct $\Delta\nu$ measurement, meaning:

- the autocorrelation showed peaks at multiples of $\sim\Delta\nu/2$,
- the folded spectrum followed the modelled template, and/or
- modes $l = 0$ and $l = 2$ aligned in the échelle diagram.

Hence, these were labelled as reliable. Meanwhile, 7,143 are labelled as unreliable either because there was absolutely no oscillation signal or signature of $\Delta\nu$, or because the $\Delta\nu$ value was considered too far off. The latter was typically the case when the $\Delta\nu$ value was offset more than $\sim$3% from the value that would align the échelle ridges. If $\Delta\nu$ is off by 3% or more the ridges in the échelle are significantly slanted (Stello et al. 2011 Figure 5), the peaks in the autocorrelation function are shifted, and the mode pattern in the folded spectrum gets slightly scrambled. The remaining 787 stars could not be confidently classified and were left out from training. Examples of stars that we visually classified as having reliable $\Delta\nu$ are shown in Figure 1. In Appendix B we show a larger sample of reliable and unreliable $\Delta\nu$.

The $\nu_{\rm max}$ distribution of stars in the training set is shown in Figure 3a. The fraction of reliable detections as obtained from the described visual method over the totals as a function of $\nu_{\rm max}$ is shown in Figure 3b. For stars with $\nu_{\rm max}$ below 10$\mu$Hz the frequency resolution of K2 data makes it difficult to measure and confidently verify $\Delta\nu$, explaining the lower prevalence of reliable $\Delta\nu$ detections. The lower detection fraction around 30$\mu$Hz, where RC stars are typically found, may be caused by their spectra showing a lower height to background ratio (Mosser et al. 2012, Equation 6) and/or due to their larger number of detectable mixed modes making the spectrum more complex (Grosjean et al. 2014, Figure 7) and hence in both cases harder for the pipelines to find the correct $\Delta\nu$ and for us to verify it. We also see a decline in the fraction of reliable $\Delta\nu$ when $\nu_{\rm max}$ approaches 100$\mu$Hz and beyond. This can be caused by the increased presence of mixed modes throughout the power spectrum and the fact that modes oscillate at lower amplitudes in this $\nu_{\rm max}$ range leading to lower signal-to-noise ratios. In the last frequency bin, when $\nu_{\rm max}$ approaches the Nyquist frequency, we can expect reflections from frequencies greater than Nyquist to interfere with the oscillation pattern, making it harder to identify good $\Delta\nu$ measurements. The shape of the histogram in Figure 3b is similar to that of the six independent pipelines analysed by Zinn et al. 2021 (Figure 12). This suggests there is little or no bias unique to our method in the training.

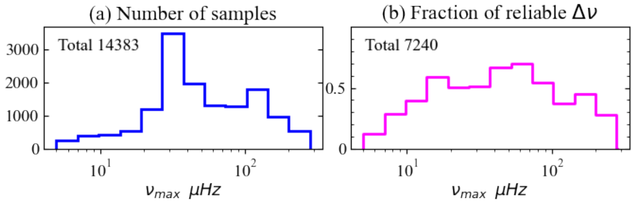

**Figure 3.** (a) $\nu_{\rm max}$ distribution of the full training set, (b) Fraction of reliable $\Delta\nu$ over the full training set.

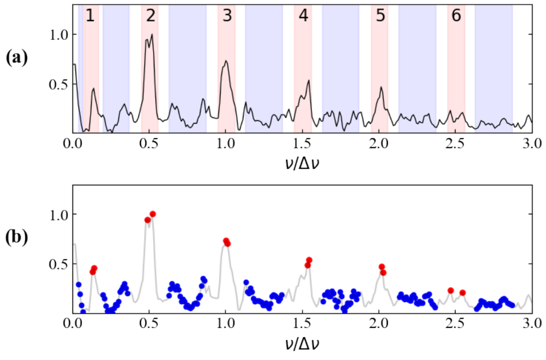

**Figure 4.** Example of calculating a reliability score from the autocorrelation function of EPIC 201207669. a) The detrended autocorrelation of the power spectrum is shown. Annotated red shaded areas mark the six regions of interest, where the autocorrelation is expected to be strongest, and the blue shaded regions where the autocorrelation is expected to be weaker. b) The final score is obtained as the weighted sum of the local contrast scores, which are calculated for each region of interest as the average of the two highest values in each red region (red dots in the figure) divided by the average of all the points inside the two blue bracketing regions (blue dots).

## 3.2 Feature Selection

The selection of input features is one of the key concepts in machine learning because the performance of the final model heavily depends on it. From our diagnostic plots described in Section 3.4 we derive four informative features to provide as input to our machine learning algorithm that mimic what we used for the visual classification of the training set. In this section we describe how we derived each feature.

### 3.2.1 Feature AC - based on the Autocorrelation Function

To automate the process of looking for the characteristic peaks in the autocorrelation function we assign a score to each autocorrelation based on the contrast between the regions where strong correlation is expected and the rest of the function. The autocorrelation function of star EPIC 201207669 (row II in Figure1b) is used to exemplify this scoring method. We treat the autocorrelation as described in Section 3.1.1, but additionally we now fit and remove its background slope using a RANSAC regressor (Fischler & Bolles 1981). In Figure 4a we show the resulting function where red shaded areas indicate our six regions of interest: The first one at $\nu/\Delta\nu = 0.12$ indicates the point of expected high autocorrelation from the pairs of $l = 0$ and $l = 2$ (Bedding et al. 2010). The rest, at $\nu/\Delta\nu = 0.5, 1.0, 1.5, 2.0$ and 2.5, indicate the expected peaks from the correlations between $l = 0$ and $l = 1$ modes. The blue shaded regions to the left and right of each red region are where we expect low correlation, and will be used to derive the contrast. Figure 4b shows, for each of the six red regions, two points with the highest value. We derive the contrast score for each red region by dividing the average of the red points by the average of the bracketing blue points.

The final AC score for the star is a weighted sum of these results with weights $w = \{0.05, 0.30, 0.50, 0.05, 0.10, 0.00\}$ chosen manually to emphasise the presence of correlation peaks at 0.5 and 1 $\nu/\Delta\nu$, but also to account for correlations at 0.12, 1.5, and 2 $\nu/\Delta\nu$, and adjusted by looking at the performance of different sets of weights for stars with good and bad $\Delta\nu$. The weights are modified to $w = \{0., 0., 1., 0., 0., 0.\}$ for low $\Delta\nu$ stars ($\Delta\nu$<1.25$\mu$Hz). For these stars we are only interested in the autocorrelation in region 3, because peaks at $\Delta\nu/2$ are no longer expected and the pair $l = 0, 2$ is no longer located at $0.12\nu/\Delta\nu$ because the oscillations pattern begins to resemble a triplet structure (Stello et al. 2014).

### 3.2.2 *Features XC1 and XC2 - based on the Folded Spectrum*

We were able to craft a good indicator of the similarity between a star's folded spectrum and its corresponding modelled template from their cross-correlation function. We call this metric XC1, and it measures how the maximum correlation coefficient compares to the rest of the correlation function across all shifts. The procedure to calculate XC1 is illustrated for EPIC 201207669 in Appendix C1.

We also tried an alternative way of quantifying the similarity between folded spectra and templates: metric XC2 is obtained by calculating the *Manhattan distance* between the model template shifted to the position of maximum correlation and a smoothed version of the star's folded spectrum. The smoothing is done applying a Butterworth filter of order 4 and cut-off frequency 18 cycles per $\Delta\nu$, and scaling the result between 0 and 1. The *Manhattan distance* is the sum of the absolute differences between the observed and modelled folded spectra at each point.

Even when both XC1 and XC2 aim to extract similar information from the data, we keep them both as inputs for the neural network because a combination of both features is a better $\Delta\nu$ reliability indicator than any one of them. (See analysis in Appendix C2).

### 3.2.3 *Categorical Feature based on $\nu_{\max}$*

There is generally some dependency of how the autocorrelation function looks with $\nu_{\max}$ and therefore adding in $\nu_{\max}$ as a feature is informative. In practice we do this by encoding $\nu_{\max}$ of each star into a categorical feature. The feature is set to define six bins of equal width in logarithmic space between 7.7 and 280 $\mu$Hz, where a $\nu_{\max}$ corresponding to the first bin generates the array $[1, 0, 0, 0, 0, 0]$, and so on.

Finally, the $\nu_{\max}$ categorical variable is concatenated with the three features AC, XC1 and XC2 into an array of length 9 for each star. This array will be referred to as *Input A*.

### 3.2.4 *Échelle Diagram as Image Input*

Like in the visual labelling process, we found that the neural network performed better when adding the échelle diagram as a feature. The validation accuracy during training tests went from roughly 91% to 94%.

To create the échelle diagrams for the network we used a $11\Delta\nu$-wide segment of spectrum around $\nu_{\max}$. To process this feature as an image we use a convolutional neural network, which is a special class of deep neural networks particularly useful for image analysis. Because échelle diagrams need to be standardised in size before they can be fed to the algorithm, we resize them into 11x150 images using nearest neighbour interpolation. Similarly to the autocorrelation function, the number of columns of the standardised diagram, 150, was chosen conservatively to not introduce smoothing that is too severe for even the narrowest peaks. Our choice of using 11 rows, or 11 $\Delta\nu$ segments, ensures that all the excess power is always encapsulated in the diagram with at least one row with little or no power on either side of the excess. The resulting images form what we call *Input B* for our neural network algorithm.

While the échelle diagram implicitly carries all the information described by the other metrics, using it together with the AC, XC1 and XC2 metrics yields the best network performance.

## 3.3 Network Architecture

There is no machine learning architecture that is a priori guaranteed to work better than other for any dataset (Wolpert 1996), therefore the only way to find the optimal model for a problem would be to evaluate them all. In practice, the way to choose a suitable algorithm for a problem is to make reasonable assumptions about the data and evaluate only a few reasonable models with a limited number of hyperparameters known to work well for similar tasks. For our classification problem we assume that a deep neural network architecture will be appropriate for *Input A* (Branch A), while we anticipate that a convolutional network (Branch B) will be suitable for our image-like features *Input B*. Once reasonable performances are reached, fine tuning of the hyperparameters is not recommended because it can lead to over fitting and hence any improvements in training are unlikely to generalise to new data (Géron 2019).

The structure of the neural network algorithm is illustrated in Figure 5. Outputs from the two branches, A and B, are merged by a concatenation layer and fed to the interpretation stage. Its role is to calculate the relative importance of the results of each branch, which becomes specially important if the two branches return conflicting outputs. The activation function used until this point in dense and convolutional layers is the Rectified Linear Unit (ReLU) defined as $f(x) = max(0, x)$, where $x$ represents the inputs. After the interpretation stage we apply a dropout layer (Srivastava et al. 2014) with a rate of 0.75. The role of this layer is to randomly and temporarily deactivate 75% of the neurons from the previous layers during each training step forcing the network to train with a different subset of neurons each time. This is done to prevent overfitting of the training set, which would otherwise happen when the model is optimised for performance on the samples that it has "seen", instead of optimising to generalise on unseen data. Note that the dropout layer is only active during the training phase of the network. Outputs from the dropout layer go to the output layer, where a single output neuron with a sigmoid activation function gives the final probability. The sigmoid function is used generally as an activation function in binary classifiers because it constraints the results to values between 0 and 1, with intermediate values (e.g., 0.5) indicating an uncertain decision.

**Table 2.** Neural Network Hyperparameters. We specify the types of layers and the hyperparameters used when defining each layer. "NN" indicates the number of neurons, "Activ." indicates the activation function used.

| Dense Branch A | | | | Convolutional Branch B | | | | | | Interpretation Stage | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Layer | Size | NN | Activ. | Layer | Size | Filters | Kernel | Stride | Activ. | Layer | NN | Rate | Activ. |
| Input | [9,1] | - | - | Input | [11,150,1] | - | - | - | - | Concat. | - | - | - |
| Dense | | [64] | ReLU | Conv2D | - | [16] | [3,7] | [1,1] | ReLU | Dense | [64] | - | ReLU |
| Dense | | [32] | ReLU | Conv2D | - | [16] | [3,7] | [1,1] | ReLU | Dense | [32] | - | ReLU |
| Flatten | - | - | - | Max-Pooling | [1,7] | - | - | - | - | Dropout | - | 0.75 | - |
| | | | | Conv2D | - | [16] | [3,19] | [1,1] | ReLU | Dense | [1] | - | Sigmoid |
| | | | | Conv2D | - | [16] | [3,19] | [1,1] | ReLU | | | | |
| | | | | Global Avg-Pooling | - | - | - | - | - | | | | |

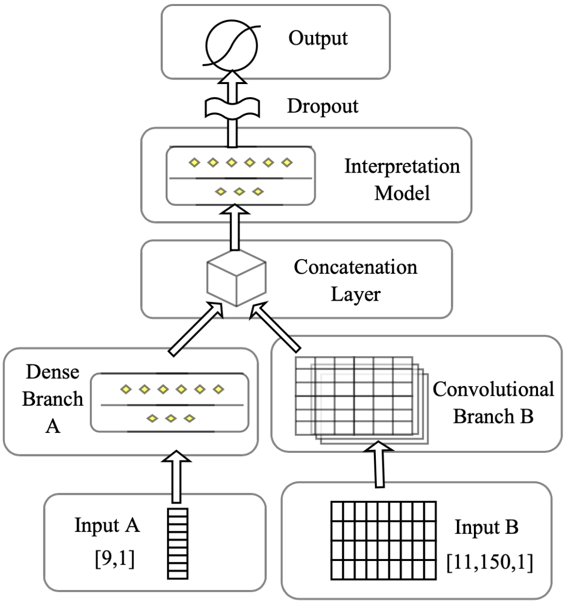

**Figure 5.** Schematic representation of the Multiple-input algorithm. Input A goes trough a fully-connected neural network (Branch A) and comes out as a 1D array of size [288], and Input B comes out of the convolutional Branch B as a 1D array of size [16], as implied by the parameters detailed in Table 2. They are concatenated before going through another fully-connected neural network, whose output is of size 1.

The hyperparameters of this neural network are summarised in Table 2. They were a design choice made by manual tuning of multiple combinations of numbers of neurons, filters, dropout rates and kernel sizes. Broadly, we followed three guidelines to reach this set of hyperparameters: (a) We required enough complexity (number of neurons) so that the algorithm would converge to a solution. (b) We needed to introduce enough regularisation so that we could train to convergence but before overfitting. (c) In Branch B, the kernels in the first couple of convolutional layers were tuned roughly to the size of the features we expect to find in Input B.

Readers interested in learning more about artificial and convolutional neural networks are referred to Appendices A1 and A2.

### 3.4 Training

From the 14,383 labelled stars, we use 75% to train the algorithm, and the remaining 25% to validate the algorithm's performance. The fraction of reliable $\Delta\nu$ (class "1") to unreliable $\Delta\nu$ (class "0") is the same in the training and validation samples.

During training we monitor the accuracy and minimise the binary cross entropy, given by:

$$-\frac{1}{N}\sum_{n=1}^{N} y_i \cdot \log(P(y_i) + (1 - y_i) \cdot \log(1 - P(y_i)) \qquad (2)$$

where $y_i$ is the truth label for each star $i$; $P(y_i)$ is the probability assigned to star $i$ by the network, and $N$ is the number of training samples. The training is done using a variant of the stochastic gradient descent algorithm called "Adam" (Kingma & Ba 2017), with a learning rate $\eta$ fixed at 0.001. "Adam" aims to minimise the loss function (which is indicative of the error rate) by adjusting the weights of the model iteratively, with the caveat that each step of the gradient descent does not use every example like the stochastic gradient descent does. Instead a mini-batch is randomly selected to train the model in steps. The size of the mini-batch determines how many steps are required to train the model using the entire training sample. One pass through the entire sample constitutes 1 epoch. The model was trained using a mini-batch size of 150.

We trained the algorithm several times. For every training session a new random data split was made and the weights were initiated at random values each time. We terminated the training just before it started to overfit. From each training session we saved the weights from the epoch with the best validation performance, and used them to build an ensemble of 39 models where each of them has a validation accuracy of ~94%. The final results of our neural network classifier are given by the average across this ensemble. This re-sampling method is known as Monte Carlo cross-validation or repeated training/test splits, and is done to allow for a better use of the limited labelled data while allowing better predictions of how well the model will perform on future samples (Kuhn & Johnson 2013).

### 3.5 Network Performance

Our choice to use cross-validation during training implies that there is no one unique validation sample unseen by the algorithm, but this also means that we have simulated a different training distribution in each training session, thus allowing reasonable estimates of model performance on unknown data using the training set.

Figure 6a, shows the distribution of the predictions made by the ensemble on the 14,383 labelled stars from the full training sample. Figure 6b is the confusion matrix when a threshold is set at t = 0.5. Black quadrants represent the correct predictions. The upper-left quadrant shows the number of true negatives, with predictions and truth labels of "0" ("Unreliable $\Delta\nu$"). The lower-right quadrant shows

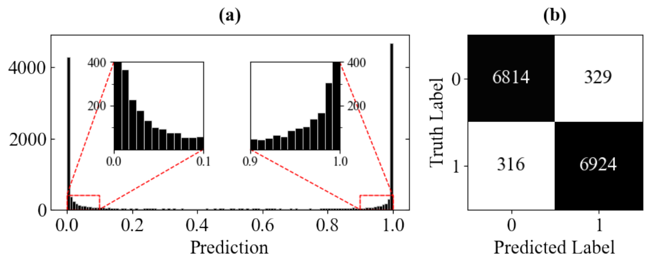

**Figure 6.** (a) Distribution of neural network predictions on the training set. (b) confusion matrix when a threshold is set at $t = 0.5$. Quadrants in black represent the number of correct predictions for each class and the white quadrants the number of mistakes for each class.

the number of true positives, the number of stars with predictions and truth labels of "1" (Reliable $\Delta\nu$). White quadrants show the number of false positives and false negatives: spectra that were predicted to have "Reliable $\Delta\nu$" (1) when the truth label was "Unreliable $\Delta\nu$" (0) and vice versa.

We use the predictions to derive the following three performance metrics: Accuracy, or the number of correct predictions made by the algorithm over all the predictions made, Precision (or purity), or the fraction of correct positives predictions among all the positive predictions made by the algorithm, and Recall (or completeness), or the fraction of correct positive predictions among all the real positives in the data set. Precision is the best metric to optimise when false positives are specifically undesirable, whereas Recall should be optimised when false negatives are specifically undesirable.

Figure 7a shows the Precision and Recall functions for different probability thresholds $t$. At probability = 0.5, the values of Precision, Recall and Accuracy reach 95.5%. This is a good decision threshold for us because for our purposes Precision and Recall are equally important.

Figure 7b shows the distribution of predicted probabilities for the mistakes made by the network. We find that most of the mistakes are indeed those with intermediate prediction values, as suggested by the fact that in Figure 7a, Precision approaches 1 for t>0.8 (meaning false positives approach zero) and Recall approaches 1 for t<0.2 (meaning false negatives approaches zero). It follows that to obtain a clean sample with highly reliable $\Delta\nu$ and a minimum number of false positives, the threshold can be set higher: for example for $t = 0.9$ Precision is 99.66%. However there would be a trade-off in Recall, which means there will be more false negatives, meaning good $\Delta\nu$ values incorrectly vetted out.

Figure 7c shows the distribution of incorrect predictions divided by total number of predictions as a function of $\nu_{\rm max}$. Green and blue represent false negative and false positive rates respectively. The red line shows the combined rate of mistakes. The greatest rates of mistakes occur around stars with $\nu_{\rm max}$=10 $\mu$Hz, which is caused by the low frequency resolution. There is also a small increase in mistakes around $\nu_{\rm max}$=30 $\mu$Hz and around $\nu_{\rm max}$=200 $\mu$Hz. We will discuss the challenges of vetting stars in these frequency ranges in section 4.1.

The results just presented tell us that when applying this neural network to a new sample we could expect our outputs to be consistent with its labelling from a human vetter ~95.5% of the time, granted that the $\nu_{\rm max}$ distribution of the new sample is similar to the one from this training set.

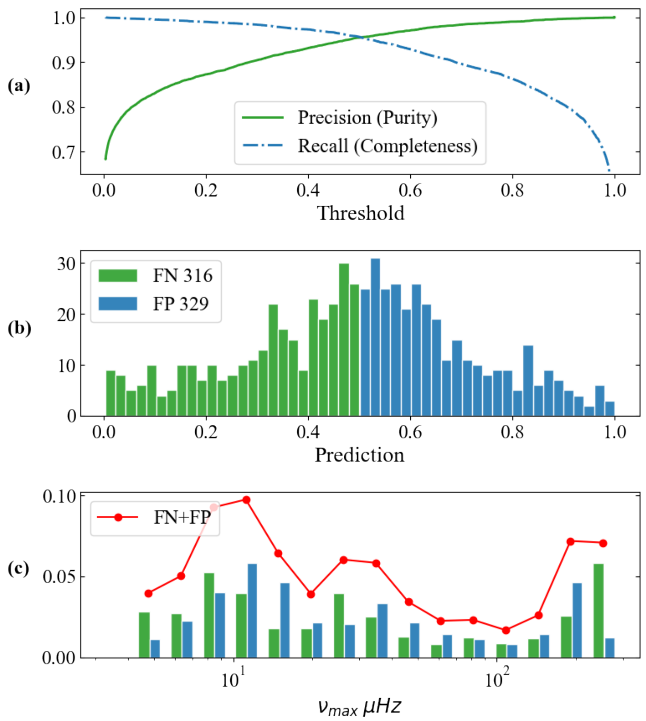

**Figure 7.** Performance of the neural network classifier on the training set. (a) Precision-Recall (or Purity-Completeness) curve. (b) Prediction distribution of mistakes. (c) $\nu_{\rm max}$ distribution of the mistakes, normalised to the number of stars from the full training set in each particular bin. In red dots the sum of False Positives and False Negatives for each $\nu_{\rm max}$ bin. Green and blue bars correspond to FP and FN (which are colour coded as in (b)).

## 4 RESULTS

In this section we present the results obtained by running our classifier on the K2 GAP sample for which $\nu_{\rm max}$ and $\Delta\nu$ are derived by different pipelines.

In order to evaluate the performance of the classifier on unlabelled data we look for agreement between the $\Delta\nu$ and $\nu_{\rm max}$ for the stars vetted by us and empirically obtained relations from well known oscillating red giant samples observed by *Kepler* (Yu et al. 2018). We do this first with $\Delta\nu$ and $\nu_{\rm max}$ predictions from the SYD pipeline followed by results from five other pipelines.

### 4.1 Results on $\Delta\nu$ from SYD pipeline

The SYD pipeline is run on all the K2 power spectra from campaigns 1-8 and 10-18 corresponding to 47,683 time series (from 45,132 unique targets) that were deemed to potentially show oscillations by the neural network detection algorithm from Hon et al. (2018b). No significance testing or other form of vetting was performed on the resulting $\nu_{\rm max}$ and $\Delta\nu$ results from this SYD run. Hence, by construction we expect a large fraction of $\Delta\nu$ values to be incorrect. Less than 20,000 stars are known to actually show oscillations with reliable seismic results for both $\nu_{\rm max}$ and $\Delta\nu$ in the K2 GAP sample (Zinn et al. 2021). The vetting method that we now implement as part of the SYD pipeline is therefore our neural network classifier, and the resulting vetted SYD values are listed in Table G1.

Figure 8a shows the $\nu_{\rm max}$ distribution for our entire K2 sample of 47,683 stars, including targets observed during more than one

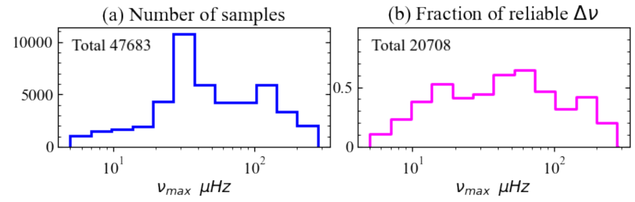

**Figure 8.** (a) Distribution of $\nu_{\rm max}$ for our full K2 sample before vetting, (b) Fraction of good $\Delta\nu$ measurements found in every $\nu_{\rm max}$ bin after vetting the sample from (a).

campaign. We can see in Figure 8b that the fraction of reliable $\Delta\nu$ measurements found by our vetting has the same general distribution as the corresponding histogram from the training set (Figure 3b).

To analyse the performance of the automated vetting, we first compare our predictions against the well established power law relation between $\nu_{\rm max}$ and $\Delta\nu$ from Stello et al. (2009), where $\Delta\nu = 0.26 \cdot \nu_{\rm max}^{0.77}$. Figure 9a shows this $\nu_{\rm max}$-$\Delta\nu$ relation with a grey line. The scatter points correspond to the entire sample of 47,683 stars colour-coded by the probability assigned to each star by the neural network. The stars in yellow, which indicate high probability of having a good $\Delta\nu$, closely follow the power law. The stars for which the classifier predicts with certainty that $\Delta\nu$ is wrong (darkest points), are those furthest from the power law; these points correspond to measurements that are unphysical. The points coloured from violet to orange show uncertainty in the predictions (interim values) and are mostly concentrated at either low or high $\nu_{\rm max}$ or around 30 $\mu$Hz, as expected from the discussion in section 3.1.4.

To further verify if our vetting performs as desired, we show in Figure 9b the relation between oscillation amplitude and $\nu_{\rm max}$, which is known to follow a power law distribution for solar like oscillators. Particularly, like seen in the *Kepler* sample (Yu et al. 2018), we see that the stars with a high probability of correct $\Delta\nu$ measurements show a sharp upper edge along the power law relation they define.

Figure 9c shows $\nu_{\rm max}^{0.75}/\Delta\nu$ as a function of $\nu_{\rm max}$, where the ordinate is essentially a proxy for mass because:

$$\frac{(\nu_{\rm max}/\nu_\odot)^{0.75}}{\Delta\nu/\Delta\nu_\odot} \simeq \left(\frac{M}{M_\odot}\right)^{0.25} \left(\frac{T_{\rm eff}}{T_{\rm eff\odot}}\right)^{-0.375}, \qquad (3)$$

and $T_{\rm eff}$ is nearly the same for all giants. The high probability points in yellow-dominated areas describe the same general shape as the *Kepler* sample shown in Figure 10. The excess of scatter points forming a vertical stripe near $\nu_{\rm max}$=46 $\mu$Hz coincide with K2's 6-hour thruster firings. The Hon et al. (2018b) method erroneously flagged these to be oscillations; Here our method can clearly identify and remove these stars whose detected signal is not astrophysical in nature. Sixty six accepted values of $\Delta\nu$ fall beyond the vertical range plotted in panel c (and d). They are mostly false positives with probabilities between 0.5 and 0.8, where $\Delta\nu$ given by the pipeline is one half of its real value, and our algorithm was misled by alignment of the wrong modes.

Finally, we show in Figure 9d the same diagram as in c but now only for stars with a vetting probability higher than 0.5, and we colour-code stars according to their RGB/RC classification from Hon et al. (2018a) based on the values of $\nu_{\rm max}$ and $\Delta\nu$ (except for stars with $\nu_{\rm max}$>110$\mu$Hz, which we all label as RGB). It is reassuring to see the similarity between Figure 9d and the corresponding diagram from the *Kepler* sample in Figure 10.

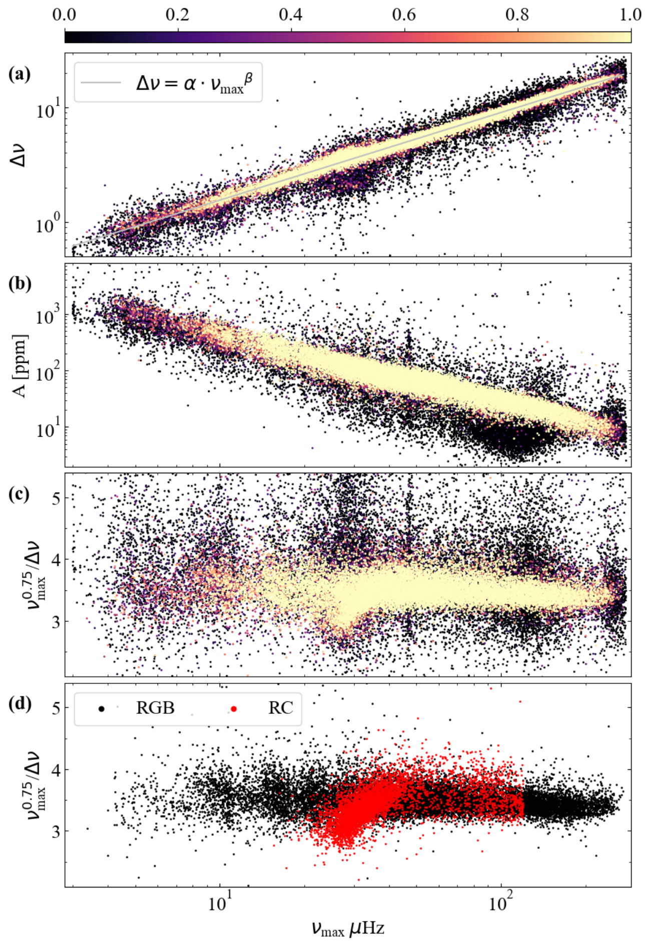

**Figure 9.** SYD pipeline results for the sample of 47,683 time series. Panel (a) shows how the SYD $\nu_{\rm max}$-$\Delta\nu$ distribution compares to the power law $\Delta\nu$=$\alpha \cdot \nu_{\rm max}^{\beta}$ where $\alpha$ = 0.26 and $\beta$ = 0.77. Panel (b) shows the distribution of oscillation amplitude as given by SYD pipeline with respect to $\nu_{\rm max}$. Panel (c) shows the distribution of the asteroseismic proxy for mass with respect to $\nu_{\rm max}$. All results on panels a, b and c are colour-coded according to the probability assigned to them by the neural network. Panel (d) shows the same distribution as in panel (c) but only for those stars with probability >0.5 (20,708 observations in total, 19,577 unique targets), and making a distinction in colour based on evolutionary phase.

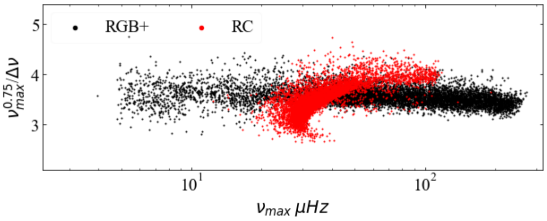

**Figure 10.** Diagram showing $\nu_{\rm max}^{0.75}/\Delta\nu$ as a function of $\nu_{\rm max}$ for the Kepler sample of 16,000 stars from Yu et al. (2018) where $\nu_{\rm max}$ and $\Delta\nu$ were derived by the SYD pipeline from 6-month spectra. All $\Delta\nu$ values were visually vetted by Yu et al. RC stars are shown in red, in black are RGB stars and stars with no RC/RGB classification available.

### 4.2 Results on $\Delta\nu$ from various pipelines

In Zinn et al. 2021 the authors perform an ensemble-based vetting on $\nu_{\rm max}$ and $\Delta\nu$ for the K2 sample as derived by six automated pipelines[1]: A2Z, BAM, BHM, CAN, COR, and SYD. Their ensemble vetting includes an iterative scaling and averaging process that makes use of results derived by the six pipelines to obtain corrected values of $\nu_{\rm max}$ and $\Delta\nu$, but in the latter case excluding A2Z. During the vetting process, $\Delta\nu$ values that deviate from the agreement of the rest of the pipelines are clipped out. The final ensemble-vetted sample contains those stars for which $\Delta\nu$ from at least two pipelines have 'survived' the clipping and contributed to the final corrected value. In this section we will be vetting the original values (unscaled and non-ensemble-vetted) used by Zinn et al. 2021 from each pipeline A2Z, BAM, BHM, CAN, and COR. Later we look for differences between our vetted samples and the ensemble-vetted sample in search for our network's mistakes. Because the SYD sample used in Zinn et al. 2021 was already vetted by our neural network, the SYD data are not treated in this section.

Our vetted results and the probabilities given by the network after running it on original samples from the five pipelines are listed in Table G2, broken down by campaign. The vetting of the samples after removing duplicated stars is summarised in Table 3a, where a distinction is made between RC and non-RC stars as per the results from Hon et al. (2018a) using ensemble-corrected values of $\nu_{\rm max}$ and $\Delta\nu$ (non-RC stars is our naming for RGB, RGB/AGB, and stars for which an evolutionary classification has not been possible). The samples used here were already vetted by the five pipelines' own internal methods. The number of those stars are listed as 'Before' meaning before our neural network vetting, while 'After' refers to after applying our vetting. For completeness, we have also included in Table 3a our vetted SYD sample from Section 4.1 after removing duplicated observations.

Figure 11 shows the stars from Table 3a before and after our vetting in the form of $\Delta\nu$ as a function of $\nu_{\rm max}$ and the mass proxy as a function of $\nu_{\rm max}$ (Equation 3) for each pipeline. RC stars appear in red, and non-RC stars in black. For all pipelines we see the vetted $\nu_{\rm max}$-$\Delta\nu$ plots (right) have been cleaned from almost all outliers compared to the left plots. In both diagrams our vetting clears out the sharp artificial cuts, which are evident in the 'Before' diagrams of all pipelines except the two Bayesian-based algorithms BAM and CAN. This cleaning reveals the astrophysical trend seen in the mass proxy vs. $\nu_{\rm max}$ diagram from Figure 10. This includes a more well-defined 'hook' of red clump stars, which is an astrophysical feature of low-mass clump stars (Huber et al. 2010, Figure 7; Mosser et al. 2012, Figure 4). However, we see a few likely incorrect $\Delta\nu$ measurements remaining after our vetting, such as the points remaining from the band under the main $\Delta\nu$-$\nu_{\rm max}$ relation for BAM around $\nu_{\rm max}{\sim}30\mu$Hz.

In Figure 12 we further investigate the Before/After neural network vetting samples for the different pipelines. The grey filled areas represent the ensemble-vetted RC and non-RC stars, which by construction is the same on the 'Before' and 'After' rows. We emphasise that the ensemble is only used for qualitative comparison with the network vetting results because the ensemble method uses corrected values of $\nu_{\rm max}$ and $\Delta\nu$ while our vetting is performed on raw values, and because the two methods are not applied to the exact same samples. Yet, our vetting appears successful at removing incorrect $\Delta\nu$ values in the case of non-RC stars, which is clear by the fact that lines representing each pipeline 'After' vetting are pushed closer towards where the ensemble-vetted results lie compared to 'Before' our vetting.

In the following we will examine the differences between our vetted samples and those from the ensemble vetting. Table 3b shows sample sizes from the different pipelines before and after the ensemble vetting. The column 'After' for each pipeline is the number of $\Delta\nu$ values that were used to obtain the final ensemble-corrected $\Delta\nu$. Table 3 shows that our method retains more non-RC stars than the ensemble method. However, our neural network is vetting out more RC stars than the ensemble method, and hence possibly removing good $\Delta\nu$ values. The lower rate of retained RC stars after our network vetting is also seen in Figure 12.

First, we look into those $\Delta\nu$ detections that our network removed but were retained by the ensemble vetting. We identify for each pipeline all the stars retained by the ensemble method for which our method returns probabilities lower than the threshold of 0.5. For many of these, the $\nu_{\rm max}$ and $\Delta\nu$ values differ by a significant amount from their respective ensemble-scaled values. Following our criteria for what we considered a good or bad $\Delta\nu$ during the labelling process, we assign as real negatives those $\Delta\nu$ departing 3% or more from the ensemble-accepted values (See Table 4, column 'RN'), while those within 3% agreement are considered suspected false negatives (Table 4 'SFN'). The rates of SFN to the total number of $\Delta\nu$ analysed from each pipeline are shown as SFN% for the total number of stars and for RC and non-RC stars separately. Figure 13 shows diagrams of $\nu_{\rm max}^{0.75}/\Delta\nu$ as a function of $\nu_{\rm max}$ for the suspected false negatives for each pipeline. Visual inspection of these stars confirmed many of them as real false negatives. We also found unclear cases where visual vetting of the spectra is ambiguous and many cases where the $\Delta\nu$ values are offset from a visually-preferred value by ~3% or more. So even though these stars are within 3% from the ensemble-accepted value, we still find quite a few of them not being accurate based on our three diagnostic plots. A sample of these 'true negatives' is presented in Appendix F. The occurrence of these true negatives was most pronounced among the RC stars. This is expected given their less precise ensemble-vetted values, which is evident from the larger spread in RC $\Delta\nu$ measurements (compared to RGB) from individual pipelines shown by Zinn et al. 2021 in their Figures 10 and 11, bottom left panels.

Moving on from stars that our vetting removed (but the ensemble-vetting did not) we now want to examine stars that our vetting preserves but that the ensemble removes. We can see such stars in the 'After' plots of non-RC stars in Figure 12, where our vetting shows a significantly larger number of accepted $\Delta\nu$ values from the BHM sample at $\nu_{\rm max}{\gtrsim}100\mu$Hz and $\Delta\nu{\gtrsim}10\mu$Hz, and from BHM and COR for $(\nu_{\rm max}^{0.75}/\Delta\nu){\gtrsim}3.5$, hence preserving stars that are taken out by the ensemble method. To further examine cases like these, we plot in Figure 14 the mass proxy diagram for all the stars found in our vetted SYD sample that were clipped out of the ensemble-vetted sample (2,862 stars in total). Visual inspection of this sample indicated that more than 97% were genuine oscillators, translating into less than 80 false positives (or less than 0.1% of the total number of stars analysed). This suggests that the ensemble vetting may be removing a significant fraction of genuine oscillators, specially at $\nu_{\rm max}{\gtrsim}100\mu$Hz. Examples of these 'extra' stars found to have reliable $\Delta\nu$ values are presented in the Appendix, Figure E1.

Overall, the analysis of our vetting against the ensemble-vetted sample does not contradict the classifier's performance metrics from Section 3.5. The four-pipeline-average of the total rates of suspected false negatives from Table 4 is 4.8%, which is slightly more than expected from our initial validation. However, it was found that not all of suspected false negatives correspond to mistakes of the network,

[1] A one-to-one analysis of the different pipelines' performances is presented in Zinn et al. 2021, Section 4.

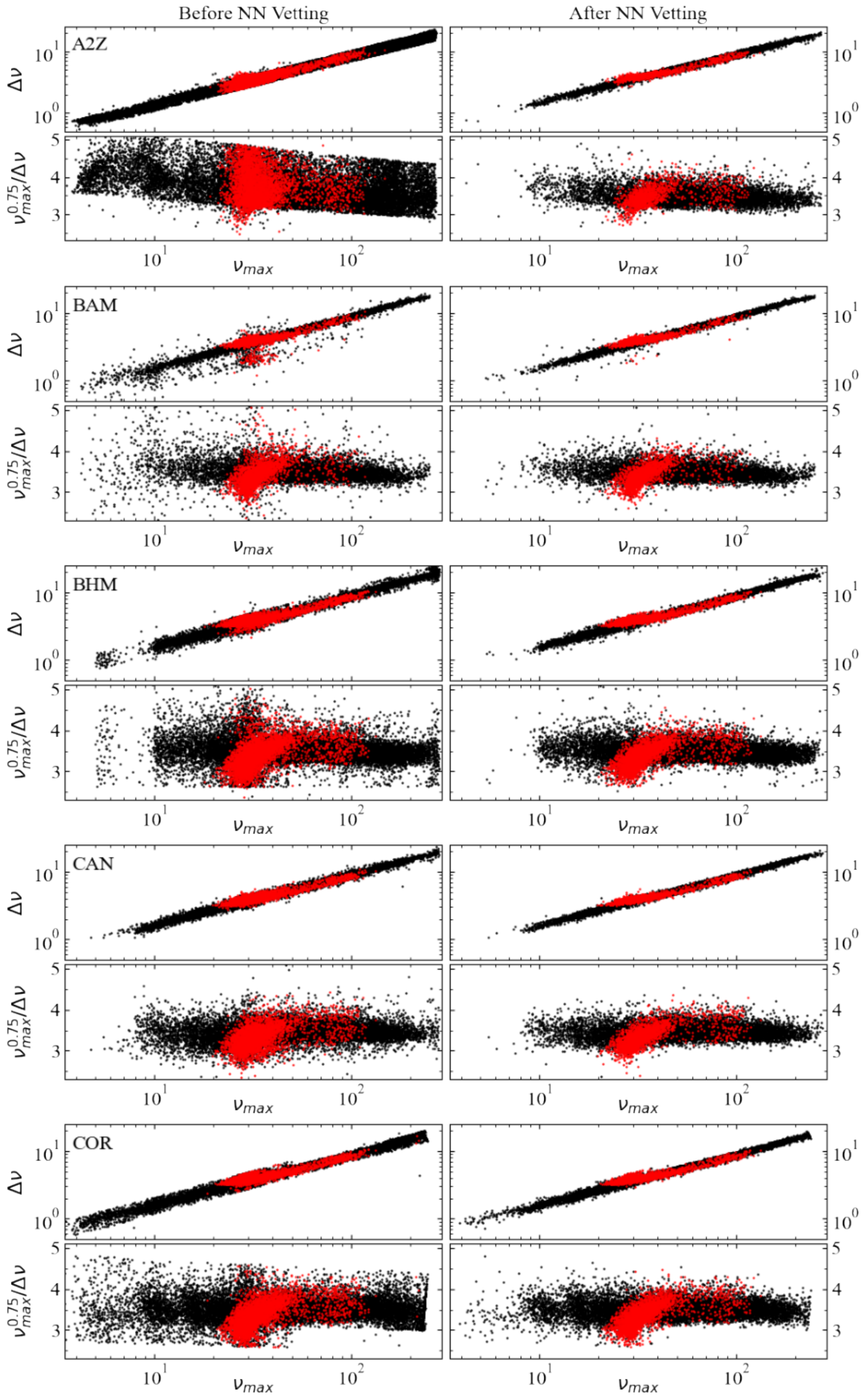

**Figure 11.** Results previously vetted by each pipeline on the left, and on the right columns we show the sample from the left vetted by our classifier. For each pipeline we show $\Delta\nu$ and $\nu_{\max}^{0.75}/\Delta\nu$ distributions as a function of $\nu_{\max}$. RC stars appear in red and non-RC stars appear in black. All frequencies given in $\mu$Hz.

**Table 3.** (a) Number of stars before and after our neural network vetting over the same samples used by the ensemble method. (b) Number of stars before and after ensemble vetting of each pipeline's results. A2Z $\Delta\nu$ values are not retained by the ensemble. The 'Before' sample sizes in (a) do not exactly match (b) because the selection process is slightly different for the neural-network and ensemble vetting. A distinction is made between Non-RC and RC stars. Total numbers are also shown.

| | Pipeline | | Non-RC | | | RC | | | Total | |
|---|---|---|---|---|---|---|---|---|---|---|
| **(a)** | Name | Before | After | Ret% | Before | After | Ret% | Before | After | Ret% |
| **N.N. Vetting** | A2Z | 18,306 | 7,653 | 41.8 | 3,869 | 1,397 | 35.9 | 22,175 | 9,050 | 40.8 |
| | BAM | 9,031 | 7,767 | 86.0 | 2,490 | 1,900 | 76.3 | 11,521 | 9,667 | 83.9 |
| | BHM | 16,456 | 12,510 | 76.0 | 5,255 | 3,527 | 67.1 | 21,711 | 16,037 | 73.9 |
| | CAN | 13,302 | 10,119 | 76.1 | 4,396 | 2,634 | 59.9 | 17,698 | 12,753 | 72.1 |
| | COR | 16,823 | 12,351 | 73.4 | 5,192 | 3,726 | 71.8 | 22,015 | 16,077 | 73.0 |
| | SYD* | - | 14,620 | - | - | 4,957 | - | - | 19,577 | - |
| | Pipeline | | Non-RC | | | RC | | | Total | |
| **(b)** | Name | Before | After | Ret% | Before | After | Ret% | Before | After | Ret% |
| **Ens. Vetting** | A2Z | 18,331 | - | - | 3,870 | - | - | 22,201 | - | - |
| | BAM | 9,421 | 7,362 | 78.1 | 2,491 | 2,261 | 90.8 | 11,912 | 9,623 | 80.8 |
| | BHM | 16,657 | 11,006 | 66.1 | 5,260 | 4,948 | 94.1 | 21,917 | 15,954 | 72.8 |
| | CAN | 13,471 | 9,512 | 70.6 | 4,397 | 4,085 | 92.9 | 17,868 | 13,597 | 76.1 |
| | COR | 18,610 | 10,985 | 59.0 | 5,197 | 4,835 | 93.0 | 23,807 | 15,820 | 66.5 |
| | ENS | - | 12,978 | - | - | 5,843 | - | - | 18,821 | - |

*Because the SYD pipeline's internal vetting used our neural network vetter, the 'Before' and 'After' numbers are the same. The number of stars for SYD are slightly different to those in Zinn et al. 2021 because the latter had additional cuts applied (see Zinn et al. for details).

**Table 4. Stars** rejected by our neural network but retained by the ensemble method for each pipeline. Columns 'RN', Real Negatives, indicate those with $\Delta\nu$ values departing 3% or more from the ensemble-scaled values. Columns 'SFN' (suspected false negatives) are those rejected having $\Delta\nu$ values within 3% of the ensemble-scaled value. The rates of SFN to the total number of non-RC, RC, and for all stars analysed by the network are shown as SFN%.

| Pipeline | | Non-RC | | | RC | | | Total | |
|---|---|---|---|---|---|---|---|---|---|
| Name | RN | SFN | SFN% | RN | SFN | SFN% | RN | SFN | SFN% |
| BAM | 238 | 82 | 0.9% | 255 | 167 | 6.7% | 493 | 249 | 2.1% |
| BHM | 432 | 375 | 2.3% | 703 | 833 | 15.8% | 1,135 | 1,208 | 5.6% |
| CAN | 375 | 226 | 1.7% | 894 | 620 | 14.1% | 1,269 | 846 | 4.8% |
| COR | 386 | 524 | 3.1% | 361 | 930 | 17.9% | 747 | 1,454 | 6.6% |

as this group also contains $\Delta\nu$ with errors of ~3% or more, and stars with unclear oscillation status upon visual verification. The real false positives, on the other hand, were found to be a fraction of a percent of the total number of stars analysed.

## 5 LIMITATIONS AND BIASES

Our classifier currently shows a higher incidence of false negatives in RC stars. This is most likely explained by our XC1 and XC2 metrics (Section 3.2.2), which are based on RGB models and thus may be too harsh in rejecting $\Delta\nu$ in RC stars. A future version of this classifier that includes templates for RC stars could help improve our vetting by reducing the rate of false negatives, thus increasing our method's accuracy.

Another limitation of our classifier stems from the difficulties in visually identifying $\Delta\nu$ reliably for certain $\nu_{\max}$ ranges. As discussed in Section 3.1.4, this limitation could be explained by the low data resolution in the case of $\nu_{\max}$ lower than ~10 $\mu$Hz, and by the lower S/N and the Nyquist frequency mirroring effects in the case of $\nu_{\max}$ higher than ~200 $\mu$Hz. We note from Figure 11 ('Before') that different methods show different efficiencies in determining $\Delta\nu$ at these $\nu_{\max}$ ranges. While most stars in these ranges are removed by our vetting, it is evident from Figure 13 that almost all of these network-removed stars were also removed by the ensemble method. This means there was no consensus on the $\Delta\nu$ value for the stars vetted out by the ensemble method, indicating that these limitations at low and high $\nu_{\max}$ are intrinsic and not a unique bias to our vetting. However, because our neural network was trained on data that was manually labelled, a 'human' bias is inevitably present.

Despite these limitations, our classifier avoids the drawbacks of vetting methods that employ sharp parameter cuts (undesirable for population analyses) and it provides an efficient way to remove outliers in $\Delta\nu$. In Appendix D we present an attempt to vet the samples using cuts to the uncertainties in $\Delta\nu$ delivered by each pipeline, which demonstrates that the uncertainties for individual stars is not a good measure to identify outliers.

## 6 CONCLUSIONS

We have presented a new automated method that efficiently vets asteroseismic $\Delta\nu$ measurements applying criteria based on the visual inspection of the spectra as defined in Section 3.1. Our automated vetting is fully independent of the method used to derive $\nu_{\max}$ and $\Delta\nu$ and does not rely on any prior knowledge of empirical relations as used by many pipelines to constrain $\Delta\nu$ detections based on $\nu_{\max}$ (Hekker et al. 2011). Furthermore, raw outputs of our classifier can

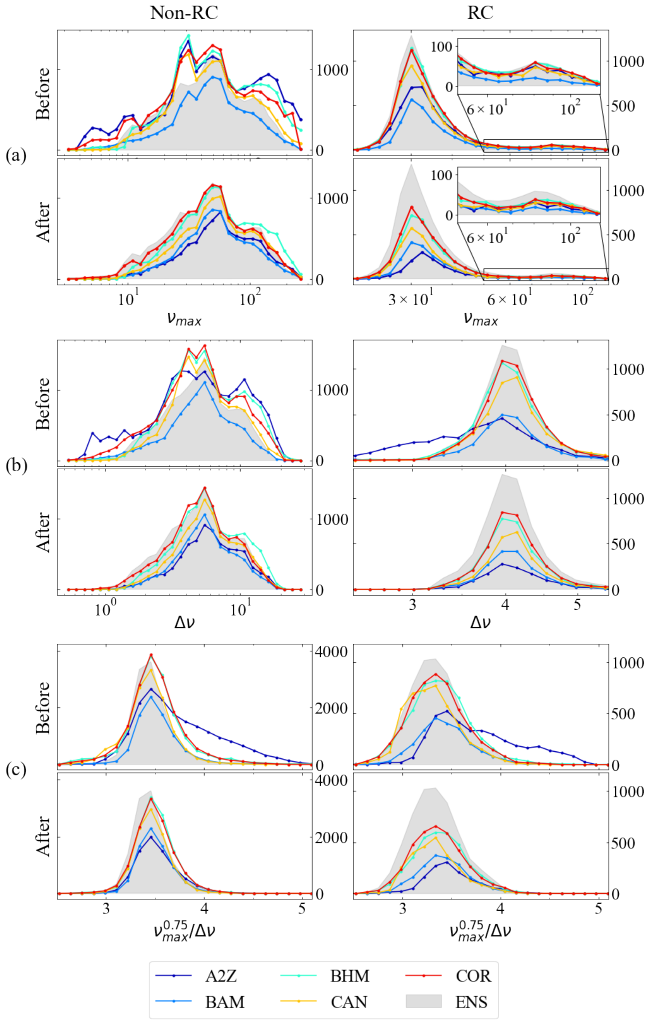

**Figure 12.** Individual pipeline histograms before and after neural network vetting showing (a) $\nu_{\max}$ (b) $\Delta\nu$ and (c) $\nu_{\max}^{0.75}/\Delta\nu$ distributions. Left column corresponds to the black points from Figure 11 and right column to the red points from the same figure.

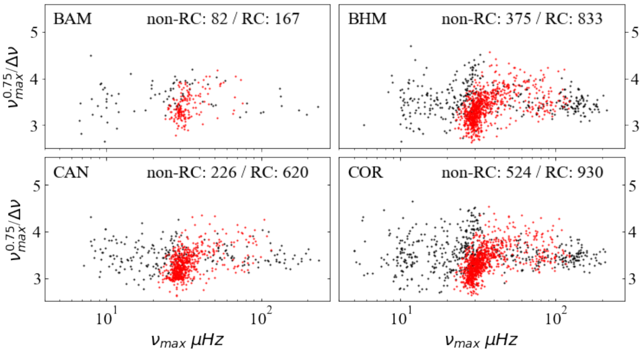

**Figure 13.** Mass proxy diagrams of the suspected false negatives from pipelines BAM, BHM, CAN and COR. From these samples we find many confirmed false negatives, but also unclear/inconclusive cases as well as cases with wrong $\nu_{\max}$ or $\Delta\nu$ (real negatives).

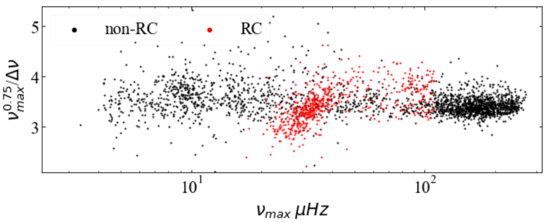

**Figure 14.** Mass proxy diagram showing SYD parameters for stars vetted by our method and clipped out of the ensemble vetted sample: 2,213 non-RC and 649 RC. These were suspected false positives, but upon visual analysis of a random sample they were found to be true positives (good $\Delta\nu$) in more than 97% of the cases.

be read as probabilities, demonstrated by the fact that hardly any of the mistakes of the network correspond to high certainty results, i.e. very close to either 0 or 1 (Figure 7b.)

From labelled training set performance we see that our neural network is expected to agree with human-vetted samples about 95% of the time, assuming the $\Delta\nu$ distributions of those samples are similar to the one used in our training.

We tested our results against trusted values from the *Kepler* sample and against K2 ensemble-vetted results. When applied to pre-vetted samples from five different pipelines we saw that the neural network removed almost all outliers from the diagrams mass-proxy vs. $\nu_{\max}$ and $\Delta\nu$ vs. $\nu_{\max}$, revealing astrophysical trends expected from the oscillation parameters of solar-like oscillations. In raw values from four pipelines we found a large number of suspected false negatives: $\Delta\nu$ values vetted out by the network but accepted by the ensemble method. Manual checks confirmed false negatives, but also revealed many $\Delta\nu$ values with errors larger than $\sim$ 3% whose rejection is by design of the training sample (Section 3.1.4). We also saw a higher incidence of false negatives from RC stars when compared to non-RC stars, which had not been previously detected. A future version of the classifier with improvements to RC performance is planned to be made available to the community, such that it could be added as a last step to any algorithm that measures $\Delta\nu$. When used on the unvetted sample from SYD we found that the neural network correctly accepted a significant number of stars that the ensemble vetting of the K2 GAP sample is discarding, especially stars with $\nu_{\max} \gtrsim 100\mu$Hz.

Overall our method appears very promising for fully automated and fast vetting of $\Delta\nu$ measurements on large samples of stars as expected from missions like TESS (as applied by Stello et al. 2021) and PLATO.

## DATA AVAILABILITY

The neural network vetted results are presented in Appendix G and are available as supplementary material.

## REFERENCES

Aerts C., Christensen-Dalsgaard J., Kurtz D. W., 2010, Asteroseismology
Baglin A., Vauclair G., COROT Team 2000, Journal of Astrophysics and Astronomy, 21, 319
Bedding T. R., 2011, Solar-like Oscillations: An Observational Perspective (arXiv:1107.1723)
Bedding T. R., et al., 2010, ApJ, 713, L176
Borucki W. J., et al., 2010, Science, 327, 977
Christensen-Dalsgaard J., 2008, Ap&SS, 316, 13
Elsworth Y., Hekker S., Basu S., Davies G. R., 2017, MNRAS, 466, 3344

Fischler M. A., Bolles R. C., 1981, Commun. ACM, 24, 381–395
Fukushima K., 2004, Biological Cybernetics, 36, 193
García R. A., et al., 2014, A&A, 568, A10
Géron A., 2019, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, https://books.google.com.au/books?id=HHetDwAAQBAJ
Grosjean M., Dupret M. A., Belkacem K., Montalban J., Samadi R., Mosser B., 2014, A&A, 572, A11
Hekker S., et al., 2011, A&A, 525, A131
Hon M., Stello D., Yu J., 2018a, MNRAS, 476, 3233
Hon M., Stello D., Zinn J. C., 2018b, ApJ, 859, 64
Howell S. B., et al., 2014, PASP, 126, 398
Huber D., Stello D., Bedding T. R., Chaplin W. J., Arentoft T., Quirion P. O., Kjeldsen H., 2009, Communications in Asteroseismology, 160, 74
Huber D., et al., 2010, ApJ, 723, 1607
Kallinger T., et al., 2010, A&A, 522, A1
Kallinger T., et al., 2012, A&A, 541, A51
Kingma D. P., Ba J., 2017, Adam: A Method for Stochastic Optimization (arXiv:1412.6980)
Koch D. G., et al., 2010, ApJ, 713, L79
Kuhn M., Johnson K., 2013, Over-Fitting and Model Tuning. Springer New York, New York, NY, pp 61–92, doi:10.1007/978-1-4614-6849-3_4, https://doi.org/10.1007/978-1-4614-6849-3_4
Mathur S., et al., 2010, A&A, 511, A46
Mosser B., Appourchaux T., 2009, A&A, 508, 877
Mosser B., et al., 2012, A&A, 537, A30
Rauer H., 2017, in EGU General Assembly Conference Abstracts. EGU General Assembly Conference Abstracts. p. 4829
Ricker G. R., et al., 2014, in Oschmann Jacobus M. J., Clampin M., Fazio G. G., MacEwen H. A., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 9143, Space Telescopes and Instrumentation 2014: Optical, Infrared, and Millimeter Wave. p. 914320 (arXiv:1406.0151), doi:10.1117/12.2063489
Sharma S., Stello D., Zinn J. C., Bland-Hawthorn J., 2021, arXiv e-prints, p. arXiv:2109.12173
Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, Journal of Machine Learning Research, 15, 1929
Stello D., Chaplin W. J., Basu S., Elsworth Y., Bedding T. R., 2009, MNRAS, 400, L80
Stello D., et al., 2011, ApJ, 739, 13
Stello D., et al., 2014, ApJ, 788, L10
Stello D., et al., 2015, ApJ, 809, L3
Stello D., et al., 2017, ApJ, 835, 83
Stello D., et al., 2021, arXiv e-prints, p. arXiv:2107.05831
Tassoul M., 1980, ApJS, 43, 469
White T. R., Bedding T. R., Stello D., Christensen-Dalsgaard J., Huber D., Kjeldsen H., 2011, ApJ, 743, 161
Wolpert D. H., 1996, Neural Computation, 8, 1341
Yu J., Huber D., Bedding T. R., Stello D., Hon M., Murphy S. J., Khanna S., 2018, ApJS, 236, 42
Zinn J. C., Stello D., Huber D., Sharma S., 2019, ApJ, 884, 107
Zinn J. C., et al., 2020, ApJS, 251, 23
Zinn J. C., et al., 2021, The K2 Galactic Archaeology Program Data Release 3: Age-abundance patterns in C1-C8, C10-C18 (arXiv:2108.05455)

## APPENDIX A: NEURAL NETWORKS

### A1 Artificial Neural Networks

An artificial neuron called a threshold logic unit is shown in Figure A1a. The threshold logic unit is a simple mathematical unit that produces an output signal by applying an activation function $\phi$ over a weighted sum of its inputs $\vec{X}$ (in our examples $\vec{X} = [x_1, x_2]$ ). Its output can be expressed as $\phi(\vec{X}^T\vec{W})$ where $\vec{W}$ represents the weights associated to each connection, which are shown graphically as arrows in Figure A1. Activation functions, also known as transfer functions, define how the weighted sum of the input is transformed into an output by mapping that sum to a predefined set of values. Examples of activation functions are shown in Figure A2. In practice, a network of neurons as shown in Figures A1b and A1c, is used to increase the complexity of functions that can be modelled or estimated with neurons.

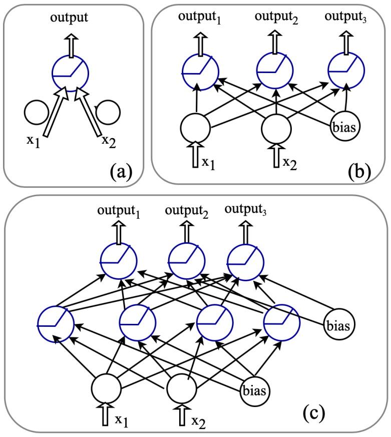

**Figure A1.** Basic structure of artificial neurons. (a) represents a threshold logic unit (TLU). (b) the perceptron architecture. (c) the multi-layer perceptron architecture MLP with one hidden central layer. Each arrow connecting neurons from different layers has a trainable weight associated (not represented here). Bias neurons represents a constant term that provides flexibility to the results of the network.

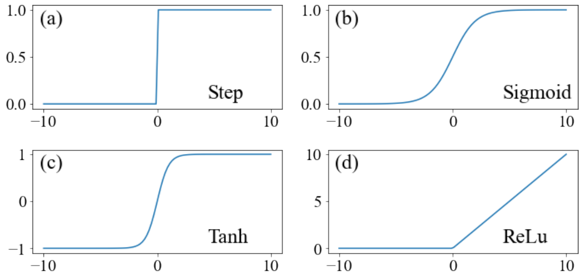

**Figure A2.** Examples of activation functions (a) A step function simply outputs '0' if its input is negative and '1' otherwise. (b) the sigmoid function: $f(x) = \frac{1}{1+e^{-x}}$, (c) Tanh: $f(x) = \frac{2}{1+e^{-2x}} - 1$ and (d) ReLU: $f(x) = 0$ if $x < 0$ and $f(x) = x$ if $x >= 0$. In our model (Figure 5) we used a sigmoid function for the output layer, and rectified linear unit (ReLU) functions in the dense layers.

The Perceptron shown in Figure A1b is a network comprising a single layer of neurons. Graphically, we can describe the neuron layer in a Perceptron as a *row* of neurons, which has connections to the neurons in layers above and/or below it but not within the row itself. Perceptrons are trained by updating their weights using the equation $w = w' + \eta(y - \hat{y})x$. During each update the connecting weights of every neuron in the network are re-calculated by adding to the current weight $w'$ the difference between the expected output $y$ and the output obtained in the previous step $\hat{y}$, multiplied by a learning rate $\eta$ and its input $x$. The weights are updated as many times as there are training instances available. This is the basis of the gradient descent algorithm, widely used in machine learning.

In practice, additional layers are stacked to form a Multi-Layer Perceptron as shown in Figure A1c. A Multi-Layer Perceptron is capable of more complex classification tasks due to the greater degree of non-linearity from more layers between input and output. Multilayer perceptrons are examples of Deep Neural Networks.

### A2 Convolutional Neural Networks

Convolutional neural networks - ConvNets - are the natural choice for machine learning tasks involving images because they are able to capture the spatial dependencies in them through the application of convolving filters.

Inspired by neurons in the visual cortex (Fukushima 2004), individual neurons in a ConvNet respond to information from only a restricted region of the input image known as their Receptive Field. The overlapping receptive fields corresponding to individual neurons cover the entire visual area. There are several elements involved when designing ConvNets: First, a *convolutional layer* made of n *feature maps*, convolves over the input layer one small sector at a time. The size of the receptive field is known as the *kernel size* as shown in Figure A3a.

All the feature maps in the same convolutional layer share the same kernel size, and all the neurons in the same feature map share the same set of weights. In ConvNets the sets of weights are called *filters* and can be represented as small images the size of the kernel that extract patterns from the inputs. Because weights in ConvNets form filters, training such network involves learning image filters that are best suited to perform a particular task.

*Pooling layers* are used as shown in Figure A3b. A pooling layer outputs a predefined function of the neurons in its receptive field called the *pool size*. Often the predefined function is the maximum value ('max-pooling') or the average value ('average-pooling').

All neural networks need to be designed according to the problem they are trying to solve. For ConvNets this involves using kernel sizes appropriate to the size of the features we wish to detect in an image. It is also important to experiment with different numbers of layers and feature map elements in order to find an optimal structure that delivers good performance while avoiding unnecessary computations.

## APPENDIX B: EXAMPLES FROM THE TRAINING SAMPLE

Figure B1 shows examples of stars with reliable $\Delta\nu$ from our training set. They represent red giants in different evolutionary phases from the bottom to the tip of the red giant branch, except for EPIC 201245474 in row VI, which appears to be a RC star because there are many mixed modes appearing all over the spectrum with heights similar to the acoustic modes. These examples are representative for our K2 sample. For high $\nu_{\max}$ frequencies (rows I and II) the presence of mixed modes is evident in diagrams b, c and d. For intermediate $\nu_{\max}$ (examples from rows III and IV) we do not see many mixed modes. A feature of our K2 sample is that occasionally some modes appear to be missing due to the stochastic nature of the oscillations and the relatively short duration of the light curves. The example in row V demonstrates this: $l = 2$ seems to be missing, but there are small peaks around the $0.5\Delta\nu$ and $1\Delta\nu$ main peaks in the autocorrelation. This indicates the quadrupoles are there, only with lower power than normal. Continuing down the list, we see that peaks appear wider; this is because the frequency resolution is the same while the frequency separation becomes smaller. The last example in row IX is one of the clearest examples of oscillations found in this low frequency range. It shows that the autocorrelation function becomes broader and that there are only a couple of orders with power. In contrast, Figure B2 shows examples of spectra with unreliable $\Delta\nu$ from our training set, which are the ones that our method aims to remove.

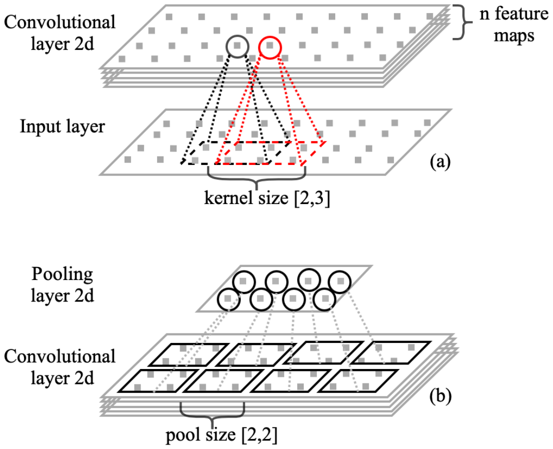

**Figure A3.** Two types of layers in a two dimensional convolutional neural network. (a) A convolutional layer of stride=[1,1]: stride is the length (in units of neurons) to skip between adjacent kernels. Here the convolutional layer has the same dimensions as its input layer, and is made of 'n' feature maps. This diagram shows these neurons do not form fully connected layers as only specific neurons of the input layer within a region that is defined by the kernel size can connect to a particular neuron in the next layer. These regions (e.g., black and red), however, can overlap. (b) A pooling layer performs an operation on groups of neurons, with the size of the group determined by a pooling size. In this example, the pool size is 2x2 and a maximum operator is applied. This means that within a group of 2x2 neurons, only the neuron with the maximum value is passed to the next layer (max pooling).

## APPENDIX C: METRICS

### C1 Calculating metric XC1

We describe the procedure used to calculate metric XC1 that quantifies the similarity between each star's folded spectrum and the folded template obtained from 1 $M_\odot$ models, as described in Section 3.1.2. Figures C1a and C1b illustrate this for EPIC 201207669: first we chose the folded model template for this star's $\Delta\nu$ of 7.9$\mu$Hz, which is model C according to Table 1. Figure C1a shows the star's folded spectrum in blue and two copies of the template in grey. Figure C1b shows the full correlation between the functions from panel (a). We create feature *XC1* by subtracting the 52nd-percentile of the correlation (green dashed line in Figure C1b) from the maximum correlation (solid green line), and divide this result by the standard deviation (green dotted line). We tried several different percentiles to calculate this indicator, and found that the 52nd-percentile resulted in the best separation of good and bad $\Delta\nu$ values in the training set. However, a similar performance could be obtained if choosing values between the 45th and the 60th percentile.

### C2 Performance of the three metrics

By using histograms we assess the individual ability of the metrics from Subsections 3.2.1 and 3.2.2 to separate reliable $\Delta\nu$ from unreliable $\Delta\nu$ in the training set of 14,383 stars. This is shown in Figure C2 where good (reliable) $\Delta\nu$ appear in magenta and bad (unreliable) $\Delta\nu$ in blue. The metric with best separability is AC because the intersection of good and bad $\Delta\nu$ distributions is only 1,796 stars.

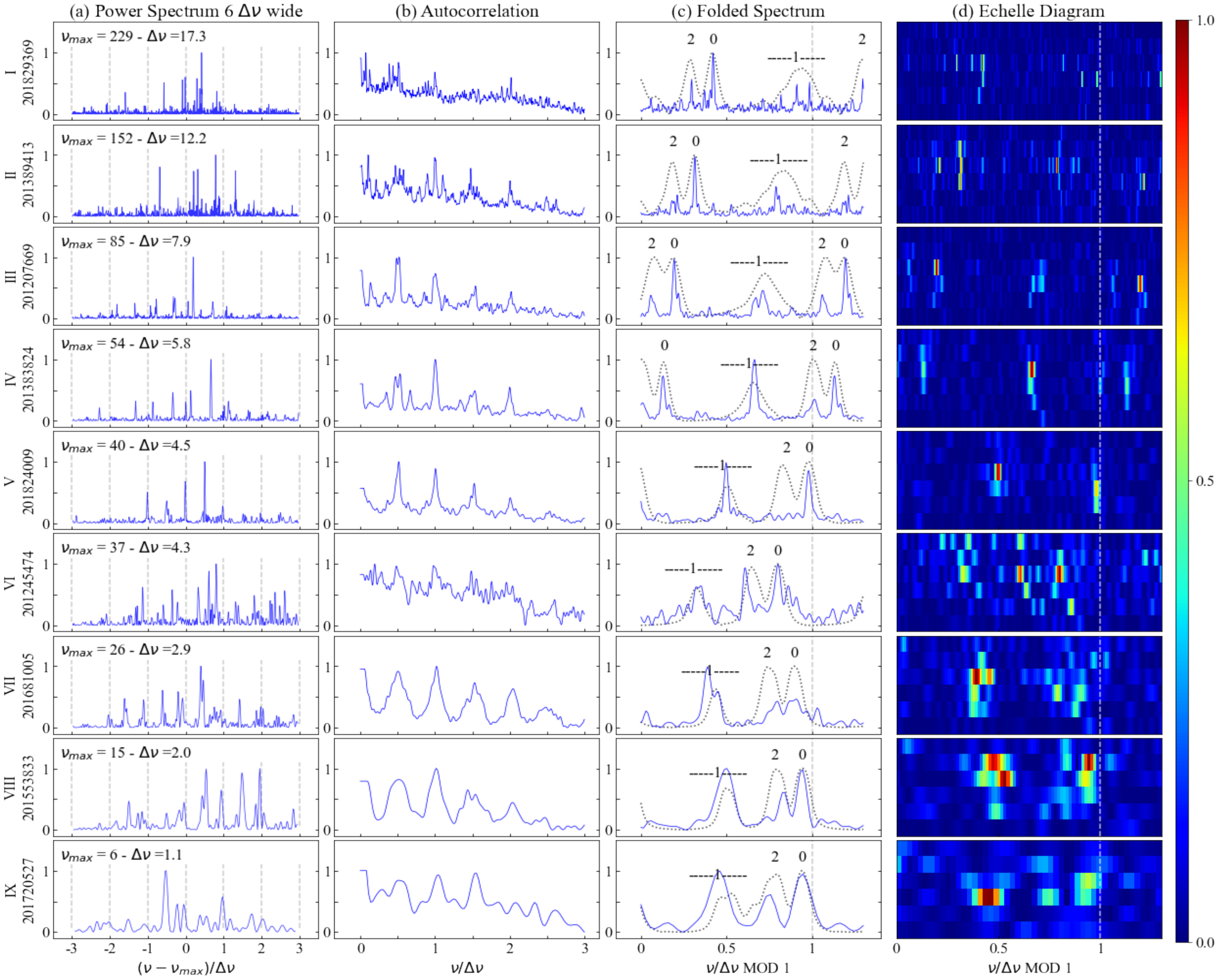

**Figure B1.** Diagnostic plots for a representative set of stars with reliable $\Delta\nu$ spanning the full range of $\Delta\nu$ values in the K2 sample. Values of $\nu_{\rm max}$ and $\Delta\nu$ are given in $\mu$Hz and all data have been scaled between 0 and 1.

This translates into 87.5% accuracy if we set a threshold at the point where both histograms (blue and magenta) have close to the same number of stars. In the same way we find that for the XC1 metric the intersection is of 2,547 stars, translating into 82.3% accuracy, and for XC2 it is 2,640 stars, leading to 81.6% accuracy.

Because the three metrics have very different numerical scales, each metric will be standardised by removing the mean and scaling to unit variance before feeding it to the neural network. Therefore, the features derived from the unseen spectra to be vetted will also be standardised to the same mean and variance from the training set.

We are now interested in testing whether the combination of pairs of metrics allows for a better degree of separability than individually. We implement a simple linear Stochastic Gradient Descent classifier and fit it on the three pairs of metrics (AC-XC1, AC-XC2, XC1-XC2) plus their labels (reliable or unreliable), represented by 1s and 0s.

Figure C3 shows the linear decision function as a green line separating reliable from unreliable $\Delta\nu$ in the two-dimensional space of each pair of metrics. Setting a threshold as in Figure C3a for the pair AC-XC1 provides an accuracy in classification of 89.2%, in (b) for the pair AC-XC2 the accuracy reaches 89.8%. Even when metrics XC1 and XC2 are based on the same characteristic of a star's power spectrum, there is indication that using both metrics is better than just selecting the best metric between them. The line separating the two populations on the scatter plot from Figure C3c still improves the individual accuracy of metrics XC1 and XC2 from 82.3% and 81.6% respectively, to better than 84% when combined. Hence, AC, XC1, and XC2 constitute good inputs for the machine learning algorithm because individually they carry non-redundant information that strongly correlates with the visual vetting or target variable.

## APPENDIX D: NEURAL NETWORK VETTING VS. UNCERTAINTY VETTING

We attempt to vet $\Delta\nu$ values from the five pipelines from Section 4.2 (A2Z, BAM, BHM, CAN and COR) using only cuts in the $\Delta\nu$ fractional uncertainty for each pipeline's measurements. Figure D1a, "Original Vetting" corresponds to the mass diagram of the same sample from the left column of Figure 11, making the same distinction between RC and non-RC stars using red and black. Columns b and c correspond to the resulting sample if we accept up to 5% and 2% of $\Delta\nu$ fractional uncertainty, respectively.

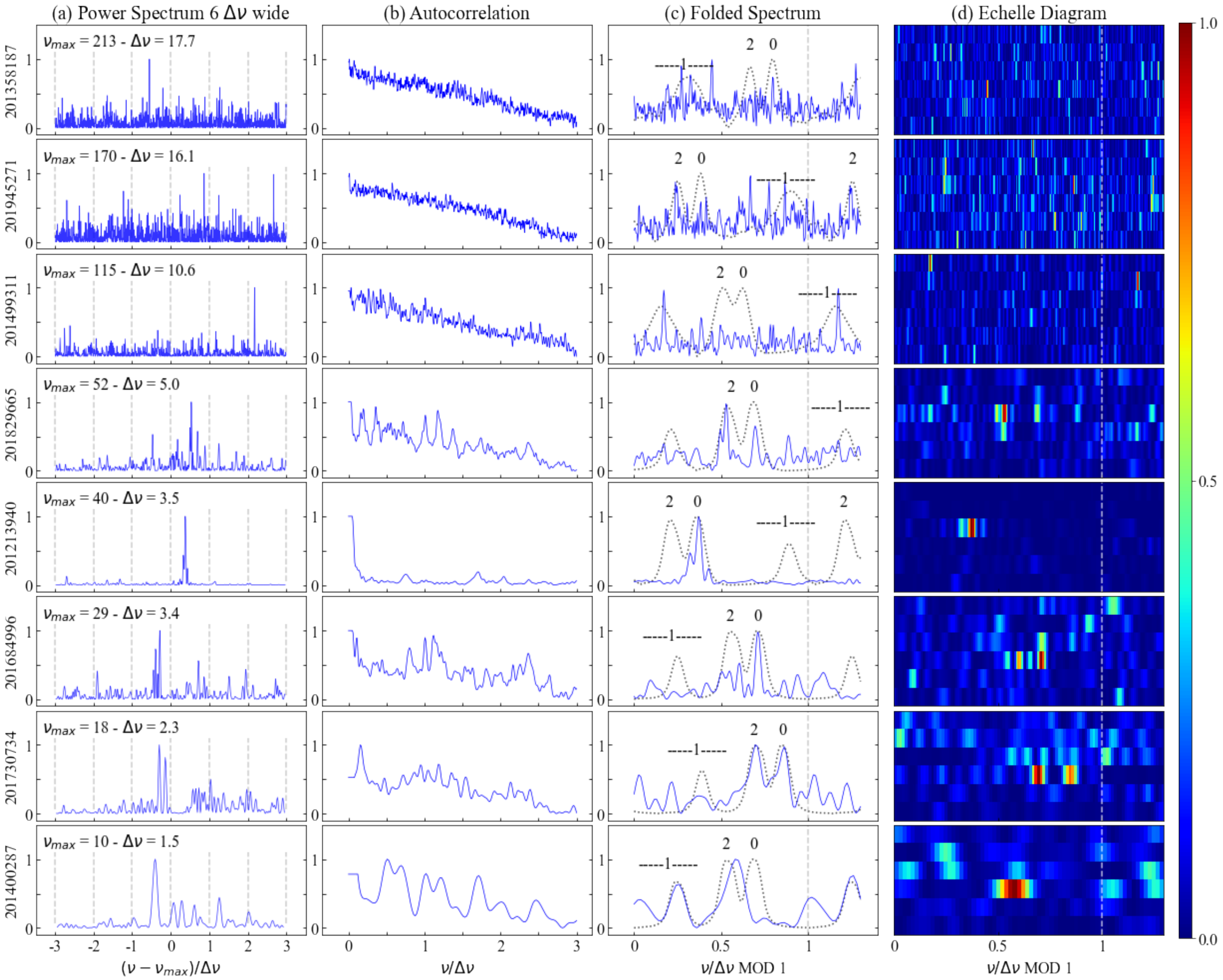

**Figure B2.** Diagnostic plots for a representative set of stars with unreliable $\Delta\nu$ spanning the full range of $\Delta\nu$ values in the K2 sample. Values of $\nu_{\rm max}$ and $\Delta\nu$ are given in $\mu$Hz and all data have been scaled between 0 and 1.

The performance of this uncertainty-vetting depends heavily on the pipeline and on the way their uncertainty is measured. For A2Z many of the clearly wrong values remain even after making the cut to fractional uncertainties lower than 2%. The CAN pipeline has very low fractional uncertainties across their entire sample, and neither of our cuts has a significant effect on it. For BAM, BHM and COR the cut to 2% does help to bring out the characteristic "hook" formed by RC stars, however too many $\Delta\nu$ values are rejected in the lower $\nu_{\rm max}$ range.

Figure D2a shows the original sample from each pipeline (same as Figure D1a) colour-coded by $\Delta\nu$ fractional uncertainty, where every point with fractional uncertainty of 5% and larger appears in black, and the lower fractional uncertainty points appear in yellow. Figure D2b uses the same colour map to show the probabilities given by our neural network classifier. Note that the colour map is inverted because we look for low fractional uncertainty in (a) and for high probability in (b). We see that our classifier performs more consistently across the different pipelines, and decidedly removes those results that are clearly outliers and brings out in yellow the known shape of the mass proxy plot for every sample.

## APPENDIX E: EXAMPLES OF APPARENT FALSE POSITIVES WHEN COMPARING TO ENSEMBLE-VETTED SAMPLE

In Figure 14 we showed the mass diagram of the stars left out by the ensemble-vetting process, but accepted by our neural network. They are mostly RGB stars with $\nu_{\rm max}$>100$\mu$Hz and most of them show clear oscillations. Here we show the diagnostic plots for four of them in Figure E1. These stars are proved to be True Positives, which is evident when looking at the folded spectra and especially the échelle diagrams.

## APPENDIX F: EXAMPLES OF APPARENT FALSE NEGATIVES WHEN COMPARING TO ENSEMBLE-VETTED SAMPLE

The analysis of $\Delta\nu$ values rejected by our classifier but accepted by the ensemble method revealed a higher number of suspected false negatives than expected from the network's measured performance, as shown in Table 4 and Figure 13. A visual check of these rejected $\Delta\nu$ values showed that this higher number can be explained by the

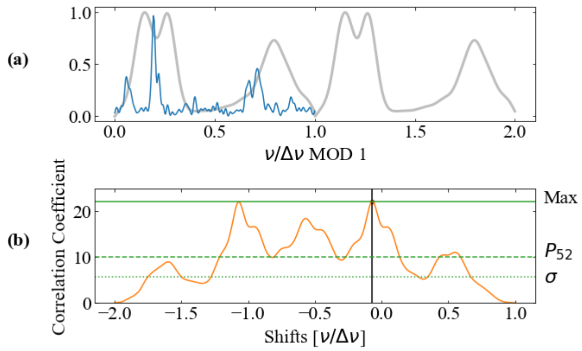

**Figure C1.** Example of calculating metric XC1 for EPIC 201207669 based on the correlation between its folded spectrum and the folded template (model C). (a) the star's folded spectrum in blue and two copies of the folded template in grey. (b) the full correlation of the functions where Shift=0 corresponds to their relative position as shown in (a). The position of maximum correlation is marked with a black vertical line. The value of the maximum correlation coefficient, the 52nd percentile of the entire function and its standard deviation are marked with solid, dashed, and dotted horizontal green lines, respectively. XC1 is then the difference between the maximum correlation coefficient and the 52nd percentile divided by the standard deviation.

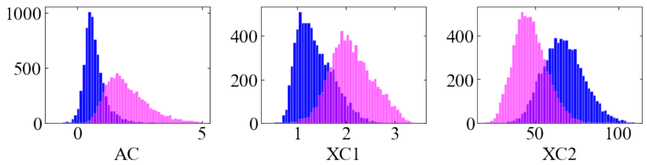

**Figure C2.** Distribution of the numerical metrics AC, XC1, and XC2 shown in magenta for stars in the training set with good Δν and in blue for the stars with bad Δν.

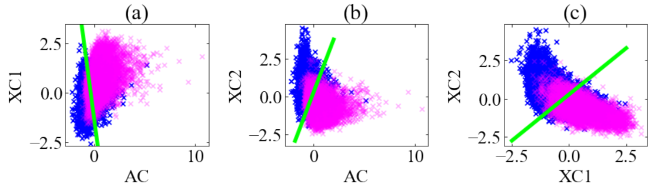

**Figure C3.** Scatter plots describing the distribution of the stars from the training set over pairs of the metrics described in Section 3.2 (after standardisation of each metric). Blue crosses correspond to unreliable Δν and magenta crosses are reliable Δν from the training set. Green lines are given by the fitting of a linear classifier to the data in the plot.

many cases where Δν is offset by ∼3% or more. These Δν values were expected rejections due to the way our training sample and the network's features were constructed. For illustration, Figure F2 shows results of manually determined Δν values found by visual inspection. This should be compared to Figure F1, which is based on raw pipeline Δν values. For diagrams (c) and (d) in particular, the manual values show much better alignment.

## APPENDIX G: TABLES

We present in Table G1 our neural network vetted results after running it on the K2 sample with SYD parameters, including duplicates. In Table G2 we present our neural network vetted results after running the network on K2 samples pre-vetted by each pipeline: A2Z, BAM, BHM, CAN and COR, using $\nu_{\rm max}$ and Δν as derived by each pipeline and including duplicates. The threshold used to discriminate the good Δν listed here was *t=0.5*.

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

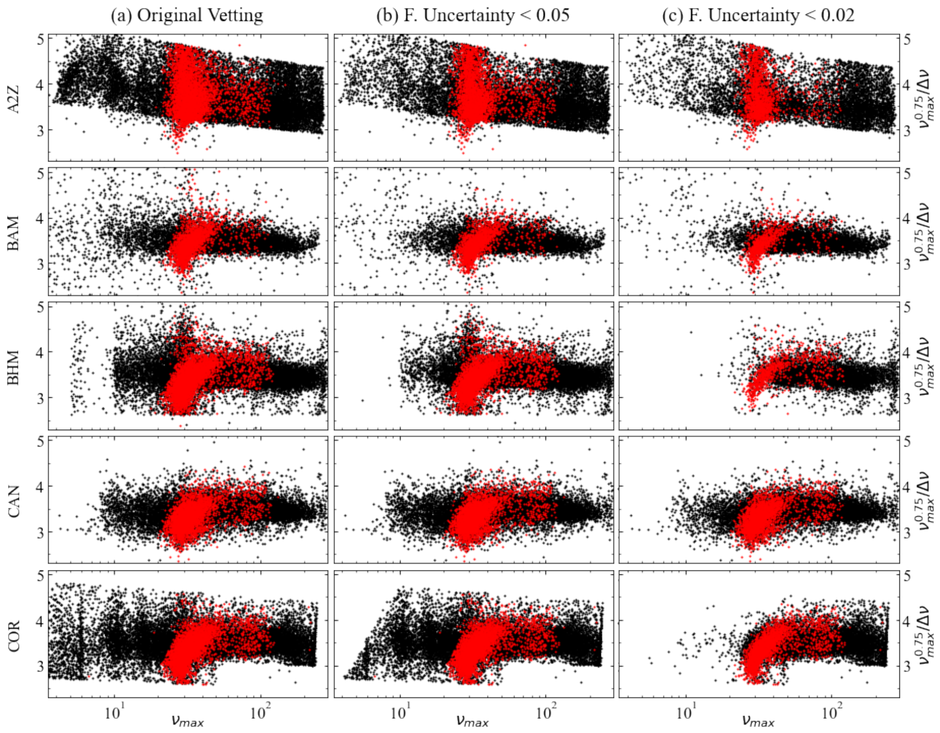

**Figure D1.** Vetting of $\Delta\nu$ using cuts based on fractional uncertainties up to 5% and 2%. RC stars appear in red and non-RC in black.

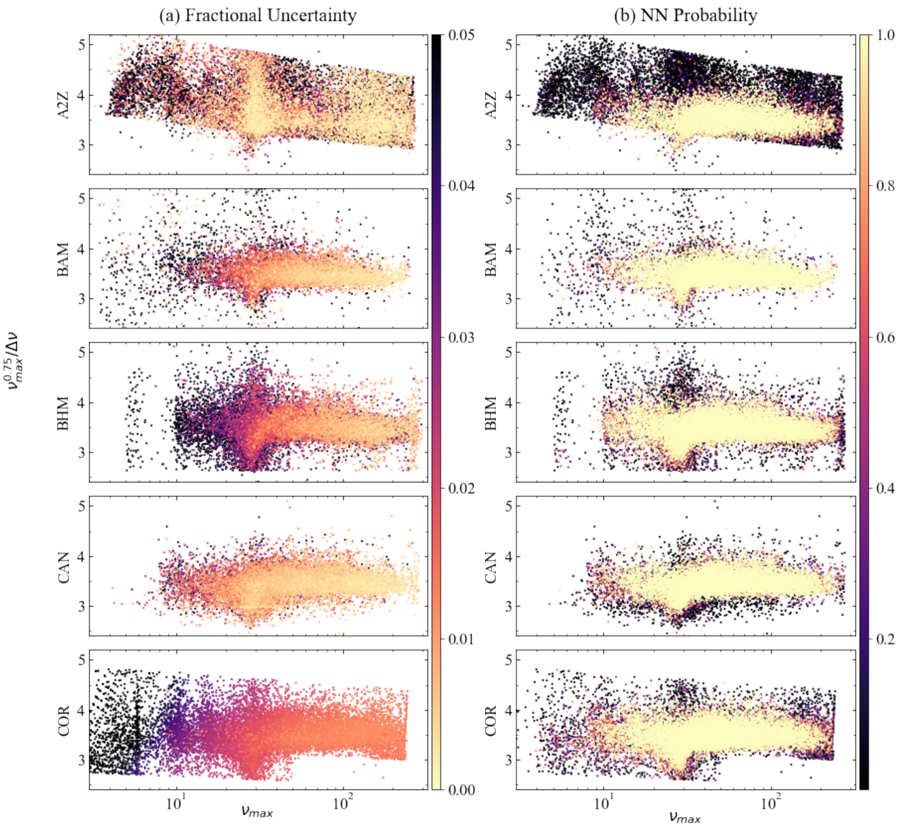

**Figure D2.** Diagrams showing $\nu_{\rm max}^{0.75}/\Delta\nu$ as a function of $\nu_{\rm max}$ colour-coded based on (a) fractional $\Delta\nu$ uncertainties as derived by each pipeline, and (b) by the probabilities given by the neural network to the values provided by each pipeline.

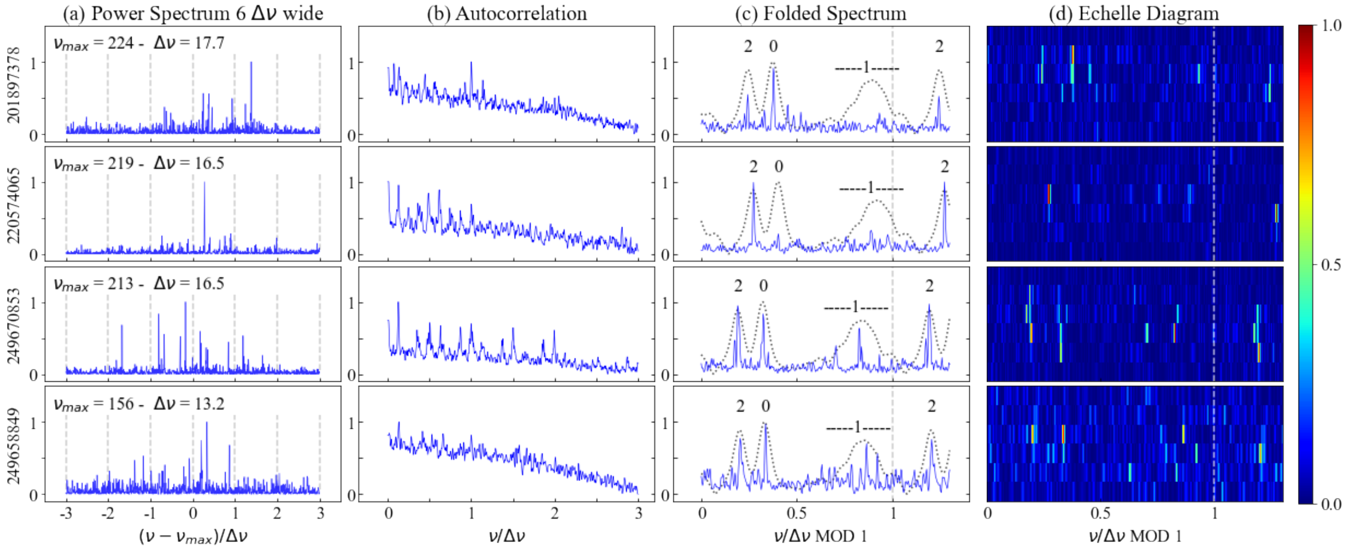

**Figure E1.** Examples of oscillating stars with $\Delta\nu$ considered good by the neural network, but missing from the ensemble-vetted sample. Values of $\nu_{\rm max}$ and $\Delta\nu$ are given in $\mu$Hz and all data have been scaled between 0 and 1.

**Table G1.** SYD results of K2 GAP sample after applying our neural network vetter. Column dnu_prob indicates the probability assigned by our neural network. RC/RGB column indicates if the star was deemed to be RGB (0) or RC (1) by the machine learning method from Hon et al. (2018a). Columns numax_sig and dnu_sig indicate the uncertainty in the result by the SYD pipeline. Values in columns numax, numax_sig, dnu and dnu_sig are given in $\mu$Hz. This table contains 20,708 observations of 19,577 unique stars. Full table available as supplementary material.

| Neural Network vetted results for values from SYD pipeline | | | | | | | |
|---|---|---|---|---|---|---|---|
| EPIC | campaign | numax | numax_sig | dnu | dnu_sig | dnu_prob | RC/RGB |
| 201670988 | 1 | 4.539 | 0.676 | 1.001 | 0.194 | 0.540 | 0 |
| 201386006 | 1 | 5.088 | 0.527 | 1.031 | 0.084 | 0.751 | 0 |
| 201135864 | 1 | 5.117 | 1.440 | 1.098 | 0.063 | 0.624 | 0 |
| 201136194 | 1 | 5.504 | 0.352 | 1.081 | 0.030 | 0.910 | 0 |
| 201364846 | 1 | 5.868 | 0.179 | 1.244 | 0.052 | 0.829 | 0 |

**Table G2.** Neural Network vetted $\Delta\nu$ values for pipelines A2Z, BAM, BHM, CAN, and COR. Values in columns numax, numax_sig, dnu, dnu_sig are given in $\mu$Hz. Columns EV_ensemble and EV indicate the evolutionary phase assigned to the star by the machine learning method from Hon et al. (2018a) for ensemble-scaled $\nu_{\rm max}$ and $\Delta\nu$ values and for values of $\nu_{\rm max}$ and $\Delta\nu$ delivered by each pipeline, respectively. Column "dnu_prob" indicates the probability assigned by our neural network. We have not removed stars with results from multiple campaigns. Full table available as supplementary material.

| Neural Network vetted results for $\Delta\nu$ values from Pipelines A2Z - BAM - BHM - CAN - COR | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Pipeline | EPIC | camp | numax | numax_sig | dnu | dnu_sig | EV_ensemble | EV | dnu_prob |
| A2Z | 201703016 | 1 | 11.032 | 0.698 | 1.880 | 0.071 | RGB/AGB | RGB/AGB | 0.990 |
| A2Z | 201727507 | 1 | 11.990 | 0.753 | 2.020 | 0.030 | RGB/AGB | RGB/AGB | 0.656 |
| A2Z | 201627037 | 1 | 12.220 | 0.786 | 2.030 | 0.325 | RGB/AGB | RGB/AGB | 1.000 |
| A2Z | 201701753 | 1 | 12.248 | 0.590 | 1.730 | 0.283 | RGB/AGB | RGB/AGB | 1.000 |
| A2Z | 201553833 | 1 | 13.400 | 2.039 | 2.020 | 0.009 | RGB/AGB | RGB/AGB | 0.994 |

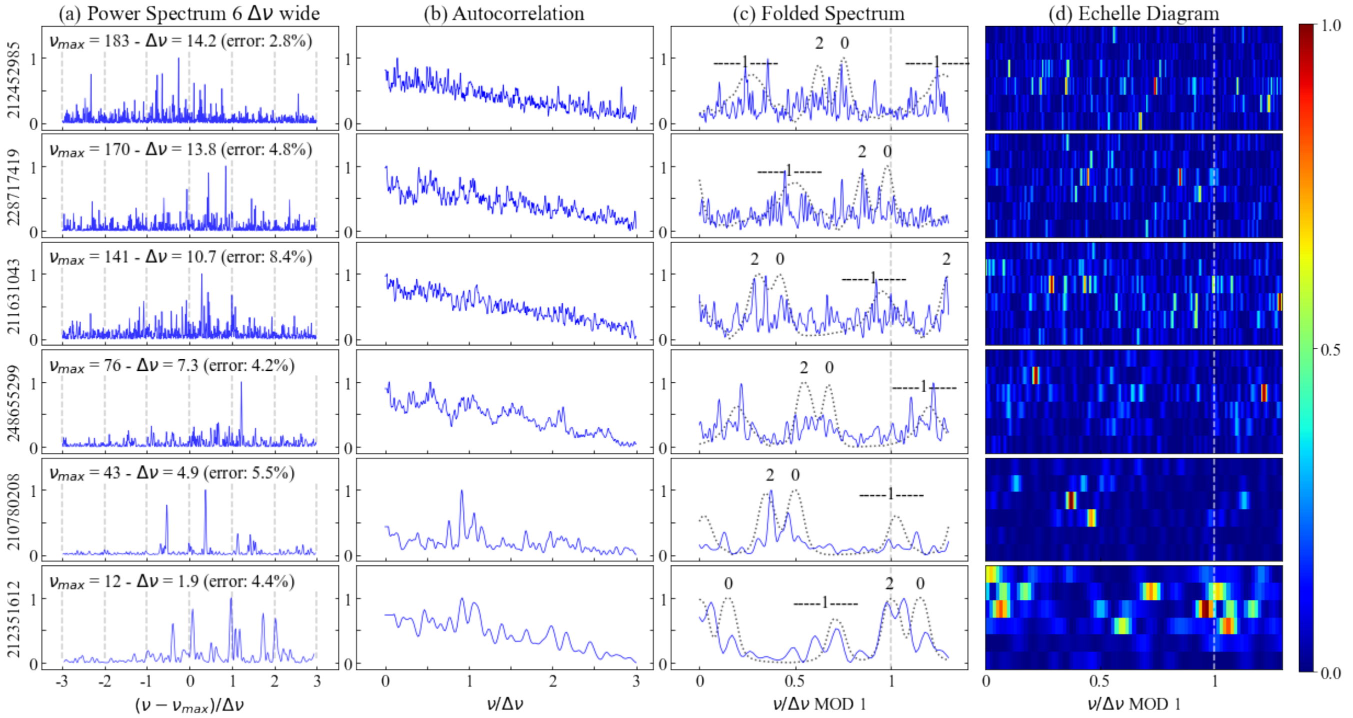

**Figure F1.** Diagnostic plots showing examples of $\Delta\nu$ from pipelines in Section 4.2 rejected by the neural network. Values of $\nu_{\rm max}$ and $\Delta\nu$ annotated in column (a) are given in $\mu$Hz and the error with respect to our manually determined value of $\Delta\nu$ is in parenthesis.

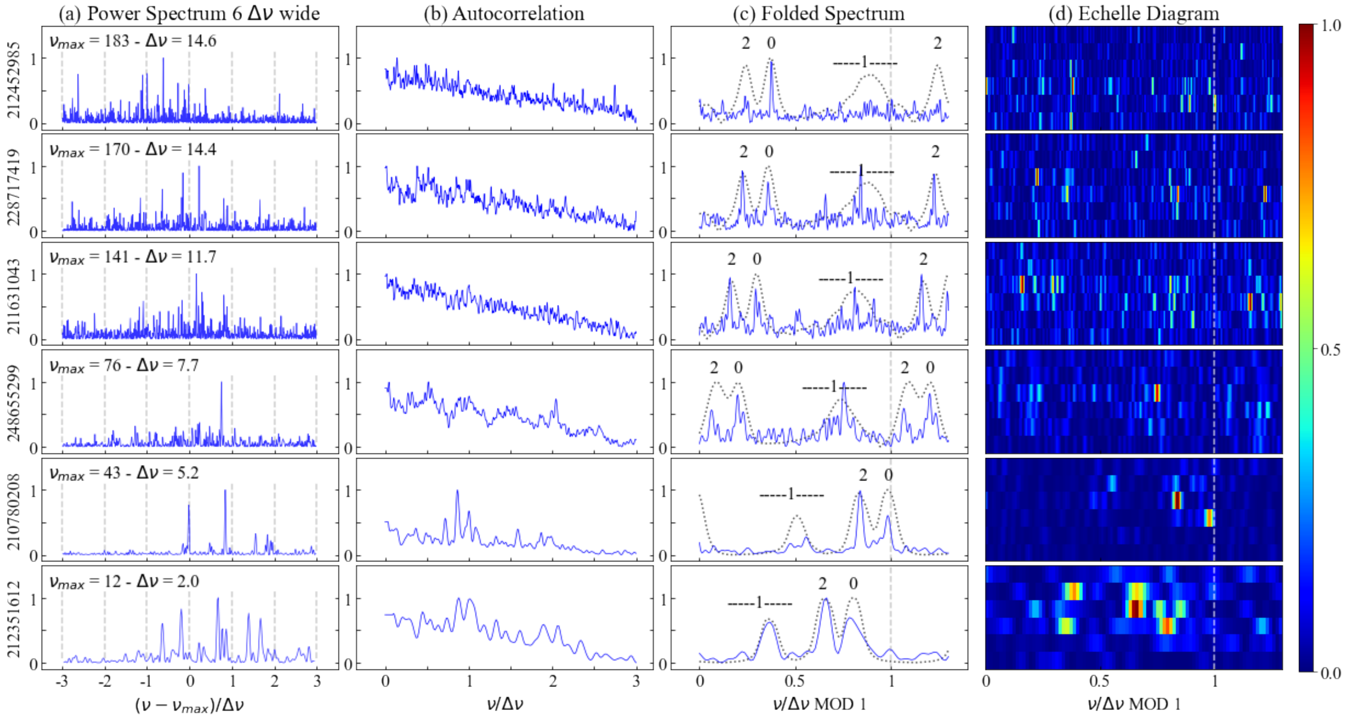

**Figure F2.** Diagnostic plots showing the same spectra from Figure F1 but with $\Delta\nu$ visually determined as the value that puts the autocorrelation peaks closer to multiples of $\Delta\nu/2$, makes the folded spectrum match the modelled template, and/or best aligns modes $l = 2$ and $l = 0$ in the échelle diagram.