Detection of Transiting Exoplanets with the *TESS* Space Mission

Student Number: 1015761

Supervisors:
Professor S. Aigrain
Dr Oscar Barragán
N. Eisner

April 22, 2020

Abstract

This project investigates the automatic classification of transits that have been detected by Planet Hunters *TESS*, a Citizen Science project dedicated to identifying transits in light curves from the *TESS* space mission. The classification accuracy of the widely used Random Forest algorithm is compared to the Probabilistic Random Forest. In contrast to previous works, this project treats transit events independently to preserve Planet Hunters *TESS*'s particular benefit in detecting single transit events. This report shows that the Probabilistic Random Forest performs no better than its classical variant, with an overall accuracy of up to 83.5% compared up to 83.6% for the random forest. It is also shown that it is possible to achieve a significant increase in performance by including multiple transit features with an overall error as low as 5.3% for the Random Forest and 4.9% for the Probabilistic Random Forest.

1 Introduction

The study of exoplanets - planets in orbit around stars other than our own Sun – is a broad and rapidly evolving field of study in Astrophysics, with the 2019 Nobel Prize in Physics being partly awarded for the first detection of an exoplanet around a solar-like star (Mayor & Queloz 1995). The discovery of 51 Pegasi b ushered in the Radial Velocity era of exoplanet detection, whereby the existence of a stellar companion is inferred from the reflex orbital motion of the star which manifests itself as a sinusoidal signal in the Doppler shift of the spectrum. This Radial Velocity method can also determine a lower bound of the companion's mass, up to a degeneracy in the unknown inclination of the system.

By contrast, we currently find ourselves in the Transit era of detection where most discoveries are by transit photometry – the observation of periodic dips in the light received from a star, caused by the partial stellar eclipse by the companion. Radial velocity measurements in this case are typically used as confirmatory follow-up. Using a combination of both methods is a powerful tool to probe the properties of the star's companion, being able to tightly constrain the mass and radius of the companion, subject to knowledge of the host star's properties. Exoplanet detection methods can reveal the abundance and nature of planetary systems in the Galaxy.

As of 12/05/2019, there have been 782 confirmed planets detected by the radial velocity method, and 3,156 confirmed detections with the transit method . The advent of the *Kepler* satellite (Borucki et al. 2010), and its repurposed mission *K2* (Howell et al. 2014), brought an explosion of exoplanet detection, boasting a combined 2,749 detections. This explosion in exoplanet detection shows little sign of slowing, with the Transiting Exoplanet Survey Satellite (*TESS*) mission (Ricker et al. 2014) aiming to find 20,000 new exoplanets and the upcoming *PLATO* mission which promises to survey up to one million stars (Rauer et al. 2014). Automatic detection of transit-like events, and the selection of the most promising candidates for radial velocity follow up is therefore rapidly becoming a critical task in efficient exoplanet detection.

1.1 Transiting Exoplanet Survey Satellite

TESS is an all sky survey dedicated to detecting transits around main sequence dwarfs with spectral classes F5 to M5, which are 10 – 100 times brighter than the stars surveyed by the *Kepler* spacecraft. It aims to maximise the potential for radial velocity follow up; there is a focus on planets with radius between that of Earth and Neptune, an under explored region of parameter space.

Its nominal two-year mission will see *TESS* employ its four optical cameras to observe the entire sky in 26 sectors, with each sector being observed for at least 27.4 days. Each star observed is assigned a unique *TESS* Input Catalogue (TIC) ID, and companions that cannot be declared false

positives by detection pipelines are designated as *TESS* Objects of Interest (TOI). Due to how the sky is divided, some stars may be observed for more than one sector.

The pre-selected target stars (numbering at least 200,000 in the nominal two-year mission) are observed with a time sampling of two minutes. In addition, full frame images (FFIs) of the entire field of view are taken every 30 minutes which can be used to construct a 30-minute cadence light curve for any star visible in the sector. These data are downlinked to Earth every 13.7 days, at perigee, therefore there will be a period around this time where no data are collected.

In its confirmed two-year extended mission (Colon 2019), *TESS* will produce 20-second cadence light curves for the preselected target stars, and FFIs taken every 10 minutes, thus allowing for even more sensitive detections .

1.2 Planet Hunters *TESS*

Planet Hunters *TESS* (hereafter PHT)[1] is a Citizen Science approach to finding transits. Hosted on the Zooniverse platform, PHT invites volunteers to manually mark where they see transits in *TESS* data. The aim is to find transits initially missed by the official *TESS* pipeline and by other teams of professional astronomers.

It follows on from its namesake Planet Hunters (Fischer et al. 2011) which analysed *Kepler* data in the manner described above. Planet Hunters showed a high ($> 85\%$) detection efficiency for planets larger than $4R_{\oplus}$ and with periods shorter than 15 days (Schwamb et al. 2012) ; Planet Hunters also demonstrated that volunteers can outperform automatic detection methods for atypical transit signals e.g. aperiodic signals from circumbinary planets (Schwamb et al. 2013) and single transit events (Wang et al. 2015). The same has shown to be true of PHT, with the detection of TOI 813 b (Eisner et al. 2020), a long (83.9 day) period planetary candidate with singular transit events in sectors 2, 5, 8, and 11. This is to date the longest period object detected by *TESS*.

As of its first anniversary, PHT has engaged over 12,000 registered users[2] and many more unregistered users who have, to date, analysed every two-minute cadence light curve. Each light curve is seen by 8 – 15 Citizen Scientists and the significance of a transit like event is evaluated by an algorithm like that described by Schwamb et al. (2012). In addition, simulated transits are injected to perform an injection recovery test to determine the accuracy of PHT. After each sector has been fully analysed by the Citizen Scientists a manual vetting process begins whereby the most significant marked events are manually analysed by a small team of professional astrophysicists to eliminate false positive readings, creating a bottleneck in the detection of planets. This project aims to develop a Machine Learning approach to the manual vetting stage of planet detection. §A shows the PHT interface as seen by the citizen scientists as well as an example of a simulated transit.

1.3 Previous automatic vetting procedures

To place this project in context, it may be useful to describe some previous automatic vetting methods. One such example is the *Kepler* Robovetter (Thompson et al. 2018). The Robovetter is a decision tree algorithm designed to designate threshold crossing events (periodic signals that cross some signal to noise threshold: hereafter TCEs) as either planetary candidates or false positives. Each TCE is presented with a series of diagnostic tests (see appendices of Thompson et al. (2018)). The threshold values for each quantitative test are chosen manually to minimise false positives whilst recovering as many planetary candidates as possible. In addition to the fact that the use of TCEs necessarily implies a periodic signal, most of the quantitative tests are carried out on phase folded light curves (the combined flux verses orbital phase for all known transits), and all TCEs with fewer than three 'good'[3] transits are automatically rejected.

Another vetting procedure is the Autovetter described by McCauliff et al. (2015). That work uses a Random Forest algorithm (see §3.2) to classify *Kepler* TCEs as either planetary candidates or false positives. The use of a Random Forest rather than a single decision tree allows, in principle, for a more robust classifier. All of the ten most significant features rely in some way on the periodic orbital solution of a planetary candidate.

A third, and slightly different, approach is that of Shallue & Vanderburg (2018) who train neural networks with various architectures on *Kepler* TCEs that have been manually classified. The neural networks are shown a whole orbital period of the (phase folded) light curve or a smaller region centered on the time of transit, or both. As with the Robovetter, TCEs with less than three events are automatically rejected. This vetting procedure also assumes no knowledge of the host star; Ansdell et al. (2018) showed that including domain knowledge has a significant increase in classification performance.

[1] https://www.zooniverse.org/projects/nora-dot-eisner/planet-hunters-tess

[2] Taken from a yet to be released document: Depper, A, Planet Hunters, Zooniverse Evaluation report

[3] i.e. not rejected by any diagnostic test.

All of these vetting procedures rely heavily on the periodic nature of transit events. Since PHT has demonstrated its ability to detect atypical transits (which may not be detected as TCEs) and, in particular single transit events, a fundamentally different approach is required for any automatic vetting.

1.4 Project overview

The remainder of this report is structured as follows. Section 2 outlines the process of candidate detection and classification; the features of interest are identified and explained. Section 3 gives an overview of Machine Learning algorithms and explains the theory of the Random Forest and Probabilistic Random Forest algorithms. Section 4 explains the metrics used to quantify the features from section 2.2. The results are presented in section 5 along with a discussion of the methods. Section 6 concludes. All code used in this project is available online [4].

2 Candidate Selection

2.1 Citizen Science approach

The first stage of finding planet candidates is for the Citizen Scientists to select where in a light curve (if indeed anywhere) there are transits. Collectively, the Citizen Scientists are presented with every two-minute cadence light curve from a given sector, with the capacity for the user to zoom into the data should they wish.

Additionally, volunteers are shown simulated light curves with a signal to noise ratio (SNR) of at least 7. The inclusion of simulated transits allows for the evaluation of the Citizen Science approach to transit detection. For registered users, an individual user weighting can be assigned based on the classification accuracy of these simulated transits.This weighting allows for better calculation of the significance of each transit-like event.

2.2 Manual Vetting

After the entire sector has been analysed, the light curves are ranked by significance and presented to a team of student and professional astronomers who eliminate false positives based on several diagnostic tests. These tests are designed to identify systematic effects and astrophysical false positives. The few candidates that survive this stage of vetting are then presented for further follow up to confirm or deny their planetary status. The diagnostic tests are explained in the remainder of this section.

[4] https://github.com/1015761/MPhys

2.2.1 Depth of Transit

To first order, a star can be considered a uniform source of light. In this regime, the fraction of light occulted by the companion is given by the fraction of the star's projected area that is obscured by the companion. By noting that the distance between star and companion is negligible compared to the distance between the system and the Earth, the relative loss of flux in transit is given by

$$\frac{\delta F}{F} = \left(\frac{R_{\text{planet}}}{R_\star}\right)^2 \tag{1}$$

Or, taking the out of transit flux F to be unity,

$$R_{\text{planet}} = \sqrt{\delta F} \times R_\star, \tag{2}$$

where $R_\star$ is the radius of the host star.

A generally accepted rule is that the maximum radius for a planet is roughly twice the radius of Jupiter, with larger objects being brown dwarfs. Consequently, the depth of the transit is one of the most important diagnostics to distinguish between planetary signals and astrophysical false positives. It is important to note that it would be impossible to constrain the companion radius without knowledge of the host stars properties.

2.2.2 Transit shape

During the periods of ingress and egress, when the companion is in the process of moving onto or away from the stellar disk, the amount of light obscured will change over time. By considering the two cases of a perfectly edge-on system in which the companion passes through the centre of the projection of the star, and the case where the companion grazes the star, one can see there is a noticeable difference in shape between the two cases. It is worth noting that, for a given $R_{\text{planet}}/R_\star$, the depth of transit for the former scenario will be larger as a larger area covers the star. In this way, a large 'v-shaped' dip can be indicative of a companion too large to be a planet.

2.2.3 Momentum Dumps

Periodically, *TESS* will undergo momentum dumps (reorientation using internal reaction wheels). Data taken within the vicinity of these momentum dumps often show anomalous fluxes which can mimic transits. Transits marked very near to momentum dumps must therefore be treated with more caution.

2.2.4 Data downlinks

TESS comes into perigee every 13.7 days, at which point it deactivates its cameras and downlinks the data to Earth. This

leaves large gaps in the data where no stars are observed. In the moments leading up to the cameras deactivating, artefacts from thermal fluctuations can cause the data to show sudden variations in flux which, if not careful, can be wrongfully interpreted as transits. As in the case of momentum dumps, therefore, more care must be taken when validating a transit very close to a data downlink.

2.2.5 Background flux

In addition to the target stars' light curves, *TESS* also records the background flux light curve. Sudden variations in the background flux can affect the light curve normalisation and mimic transit signals. Dips in the flux with almost coincident spikes in the background may therefore be due to these normalisation effects.

2.2.6 Centroid jitter

Sometimes a transit signal can be caused by the *TESS* pixel capturing light from an eclisping binary system, as well as the target star. In the case of these so-called blended eclipsing binaries, the centroid (flux weighted mean position) of the star will appear different in and out of transit. Therefore sudden variations in centroid position around the time of transit could indicate an astrophysical false positive.

2.2.7 Relative frequency of marking

It is assumed that any genuine astrophysical effects will be homogeneously distributed throughout the light curves in a sector; by looking at the distribution of marked times over all light curves in an observational sector, one can identify regions of abnormally high frequency. These have a greater chance of containing signals arising from systematic events. Such light curves are excluded before the manual vetting stage so this metric will not be used as a feature for the Machine Learning algorithms but is an important check for systematic effects nonetheless.

3 Machine Learning

Machine Learning algorithms can be broadly divided into two categories: supervised and unsupervised. A supervised algorithm learns a mapping between a training set of objects (transit events) with known features (depth, distance from momentum dumps etc.) and labels (whether to keep or reject the transit). In practice, a small subset of the training set is not fed to the algorithm to provide an unseen test set with which to measure the performance of the classifier. An unsupervised algorithm is not provided with labels and instead aims to identify clusters of objects in feature space. Armstrong et al. (2016) demonstrate an unsupervised learning approach to classification of *Kepler* and *K2* light curves. Although they report a lower overall accuracy than the supervised learning approaches, unsupervised learning can play an important role in future missions; this approach can be used from very early in the mission, owing to the fact that unsupervised learning algorithms don't require a large, manually vetted training set. Another example of an unsupervised learning algorithm in the context of this work is the DBSCAN algorithm used to combine individual Citizen Scientists markings. This project assesses the performance of two supervised learning algorithms in the classification of marked transits: Random Forest (RF) and its variant Probabilistic Random Forest (PRF). This section provides a brief theoretical background on both algorithms.

3.1 Decision Tree

Since a Random Forest is an ensemble classifier formed from individual decision trees, it will first be useful to examine the structure of an individual decision tree. A decision tree is a series of nodes, each with a condition of the form $x_j > x_{j,th}$ where x_j is one of the features of the data set and $x_{j,th}$ is a threshold value for that feature. The objects reaching the node that satisfy this condition are propagated to the right and those that don't are propagated to the left. This process continues recursively subdividing the input data set, resulting in the tree-like structure of the model.

Each node of the tree, when presented with a set of objects, chooses the feature x_j and threshold $x_{j,th}$ that results in the 'best' separation of the resulting subsets. A common definition of 'best' is the choice which minimises the Gini impurity G, defined as

$$G = 1 - \left(P_{n,A}^2 + P_{n,B}^2\right) \tag{3}$$

Where $P_{n,A}$ and $P_{n,B}$ are the fraction of the objects reaching the node n with the label A or B respectively, this can be trivially adapted to problems with more than two classes. The algorithm finds the optimal feature and threshold to find the splitting which minimises the combined impurity

$$G_{right} \times f_{right} + G_{left} \times f_{left} \tag{4}$$

Where G_{right}, G_{left} are the impurities of the groups to the right and left of the threshold respectively and f_{right}, f_{left} are the fractions of the objects reaching the node that are assigned to each group.

This process will repeat while the combined impurity of the two resultant groups is less than the impurity of the total group. The nodes for which this is not true are called terminal nodes or leaves.

By default, each terminal node will contain objects of a single class, resulting in perfect performance on the training set but unreliable performance on unseen objects. This can be somewhat mitigated by specifying the maximum depth of the tree or the maximum number of nodes.

Once the tree has been trained on the labelled training set, the unlabelled objects are passed to the topmost (root) node and propagated through the model until they reach a terminal node from which the classification is determined.

3.2 Random Forest Ensemble

A single decision tree is prone to over-fitting to the training data set and struggles to generalise to new data sets (Breiman 2001). The Random Forest is a widely used algorithm for both classification and regression problems. It is an ensemble sampler consisting of a large number of decision trees trained on a random subset of the total training set, and by using a random subset of the features for each tree;this results in trees with different conditions in their nodes and is more robust to unseen data.

Once the model has been trained, an unseen object is passed through all trees of the ensemble and the classification is a majority vote of the classifications of the individual trees. Breiman (2001) showed that a Random Forest algorithm could generalise well to previously unseen data sets. For especially large trees, the decision trees can be run in parallel, allowing for significant computational gains.

3.3 Probabilistic Random Forest

The Probabilistic Random Forest (Reis et al. 2018) is an expansion of the traditional RF algorithm to account for uncertainties in the features and labels of objects It is conceptually similar to the traditional RF with the exception that features and labels are treated as probability density functions and probability mass functions (PMFs: i.e. rather than taking simply the most probable class, giving an associated probability for each class), respectively and objects may propagate down both branches of a node with an associated probability. As such it is more versatile to noisy data sets than its deterministic counterpart.

The architecture of the PRF is similar to the RF. At each node, for a given feature and threshold $\chi_{\rm th}$, an object will propagate along both the left and right path with probability $F_{i,k}(\chi_{\rm th})$ and $1 - F_{i,k}(\chi_{\rm th})$, respectively. Here, $F_{i,k}(X)$ is the cumulative probability distribution for feature k on object i

$$F_{i,k}(X) = P(x_{i,k} \leq X) \tag{5}$$

The probability that an object i reaches a node n, $\pi_i(n)$ is therefore given by

$$\pi_i(n) = \prod_{\eta \in R} F_{i,k_\eta}(\chi_\eta) \times \prod_{\xi \in L} (1 - F_{i,k_\xi}(\chi_\xi)) \tag{6}$$

where L, R are the series of left and right turns necessary to propagate to the node n.

The PRF is trained in a qualitatively similar way to the traditional RF; features and thresholds are chosen to minimise the modified Gini impurity

$$\bar{G}_n = 1 - (\bar{P}^2_{n,A} + \bar{P}^2_{n,B}) \tag{7}$$

where the expectation value for an object at the n^{th} node to have label l is given by

$$\bar{P}_{n,l} = \frac{\sum_{i \in n} \pi_n(n) p_{i,l}}{\sum_{i \in n} \pi_n(n)}. \tag{8}$$

Here, $p_{i,l}$ is the probability that the i^{th} object has the label l.

In each decision tree, the object will arrive at multiple terminal modes with some probability and so its overall classification is no longer deterministic. Much like the traditional RF classifier, the classification of the PRF is given by a majority vote of the trees in the ensemble, where the classification from each tree is taken to be the classification with the largest probability.

Since the PRF allows objects to pass through multiple nodes, it can become computationally expensive. Reis et al. (2018) showed that ignoring nodes for which $\pi_i(n)$ is lower than some p_{th}, taken to be 5% in that work, reduced computation time with no significant decrease on classification performance.

4 Feature extraction

To apply Machine Learning to real world data, there is typically a necessary stage of feature extraction - the calculation of features from the raw data.

In each case, the script responsible for feature extraction is provided with the Flexible Image Transport System (FITS) file available from the Mikulski Archive for Space Telescopes (MAST)[5], as well as the marked times of transit from PHT. The content of the FITS files is explained in great detail in section 5 of Tenenbaum & Jenkins (2018). The data under the following headings are extracted.

- `PDCSAP_FLUX`[6] - The time series flux from the star corrected for some known systematics (e.g. pointing jitter and long-term drifts).

[5] https://archive.stsci.edu/tess

[6] Presearch Data Conditioning Simple Aperture Photometry Flux

- SAP_BKG - The estimated background flux contribution to the target.
- TIME - The Barycentric *TESS* Julian Date (BTJD)[7] time of each measurement.
- PDCSAP_FLUX_ERR - The error on each flux measurement.
- QUALITY - Bitwise flags for known systematic effects (e.g. a momentum dump occurring)
- MOM_CENTR1 and MOM_CENTR2 - The CCD row and column location of the flux weighted centroid, respectively.

The periodic nature of astrophysical events is intentionally ignored. The goal is to best approximate the human vetting process which has shown its success for single transit events so objects are treated independently. This is a different approach to previous automatic vetting procedures (see §1.3).

The following features are extracted for each light curve independently, allowing for the code to be easily parallelised. The majority of the analysis is carried out on a small segment of the light curve, typically centred on the marked time of transit. A trapezoidal transit signal with a low order polynomial multiplicative trend is used to find the best fit to the data using the Levenberg–Marquardt χ^2 minimisation algorithm provided in the `lmfit` package[8]. This was chosen as it provides a fast, robust non-linear least squares fitting algorithm which can estimate parameter uncertainties.

The density based clustering algorithm that combines the individual identifications from each volunteer can occasionally, for very short period signals, combine the markings for two separate events. Since there is no guarantee therefore that the marked times correspond to the central time of transit, the parameters free to vary are `tmid`, the centre of the transit; `dur`, the duration of transit (from beginning of ingress to end of egress); `h` the relative depth of transit; and `v` a parameter to quantify the shape of transit. The out of transit flux is constrained to be unity[9].

In order to reduce the chance of converging to a local minimum, the minimisation is run 5 times with randomly distributed starting parameters. Should none of these fits converge, the light curve has any long-term trends removed with the `wōtan` python package (Hippke et al. 2019) and the out of transit flux is constrained to be unity everywhere. If any of the following tests fail to produce a sensible value (or, indeed, any value at all), a placeholder `NaN` value is used as a flag for later analysis.

Fig. 1 outlines the feature extraction process applied to each star in each sector. The features extracted from the data set are the companion radius, the shape of transit, the time to momentum dumps and data downlinks, spikes in background flux, signal to noise ratio, and centroid jitter.

4.1 Companion radius

As seen in Eq. 2, the radius of the companion is proportional to the square root of the relative reduction in flux. The relative dip `h` is given directly by the fitting code.

For the majority of targets, the value of $R_\star$ is known and displayed at the stage of manual vetting. For the minority of targets for which the stellar radius is unknown, the stellar radius is taken to be $1R_\odot$ following the approach of McCauliff et al. (2015). Where relevant, the uncertainty on the stellar radius is also given by $1R_\odot$ to encode our almost total ignorance on the stellar parameters. The uncertainty on the depth of transit is given directly by the minimisation algorithm.

4.2 Shape of transit

As with the depth of the transit, the value of `v` and the associated error are produced directly by the fitting code. `v` is defined as the fraction of transit duration spent in either ingress or egress; `v` = 1 would produce a triangular signal, and `v` = 0 would be a rectangular signal.

4.3 Time to systematic effects

The `QUALITY` header contains, amongst other flags, the times of momentum dumps. The difference between the marked time and the nearest time of momentum dump is stored as a feature.

In addition, since the value of `PDCSAP_FLUX` for cadences when the data is being downlinked to Earth is stored as `NaN`, finding the distance from the marked time to the nearest series of 10 consecutive `NaN` values gives the time to nearest data downlink [10].

Note that for momentum dumps, the difference in times is stored, whereas only the distance to the nearest downlink is of interest; cadences directly before a downlink are treated identically to cadences directly after a downlink. In both cases,

[7] BJD - 2457000

[8] https://github.com/lmfit/lmfit-py

[9] The inclusion of the multiplicative polynomial trend acts to relax this constraint.

[10] The criterion of 10 consecutive nans is due to the fact that some bad data are removed by the *TESS* team before upload, causing sporadic `NaN` values in the data set not caused by data downlinks; the number 10 is chosen somewhat arbitrarily and shouldn't affect the remainder of this analysis.

the uncertainty on these measurements is given by the uncertainty on the time of centre of transit which is provided directly by the fitting algorithm.

4.4 Background jitter

The effect of the background spikes is quantified by the deviation from some smooth trend. First the time around transit is masked out to fit a low order polynomial to the out of transit trend. This background flux is then normalised by this trend. Secondly, the mean squared error of the masked (in transit) and unmasked (out of transit) regions of the normalised background flux are calculated. The feature extracted is the ratio of these MSE values.

Since this is a highly non-linear function of transit time, the uncertainty cannot simply be derived by propogating the uncertainties in the fitting parameters. Instead, the background jitter is calculated at the earliest and latest times allowed by the uncertainty on transit time derived from the least squares minimisation. These values are then used to estimate the uncertainty.

4.5 Signal to noise ratio

In addition to the metrics mentioned in Section 2.2, the signal to noise ratio (SNR) serves as a useful proxy for the statistical significance of any features, which will implicitly affect the confidence in identifying a genuine astrophysical event.

Strictly speaking, the SNR is defined as

$$\text{SNR} = \sqrt{N}\frac{\delta F}{\sigma_{\text{OOT}}}, \tag{9}$$

for N transits with depth δF and out of transit standard deviation σ_{OOT}. In this treatments, transits are analysed singly, so $N = 1$. As a good first order approximation, the fractional uncertainty in SNR is given by the fractional uncertainty in the depth of transit which is given directly by the fitting algorithm.

4.6 Centroid jitter

Sudden variations in the centroid (from some smooth overall trend) can mimic transit events. This jitter is quantified as

$$\hat{\sigma_r} = \sqrt{\hat{\sigma_x}^2 + \hat{\sigma_y}^2} \tag{10}$$

where the quantity $\hat{\sigma}$ is the spread of data during the times of transit expressed as a fraction of the out of transit spread. The subscripts x, y refer to the data sets `MOM_CENTR1` and `MOM_CENTR2`, respectively. The uncertainty is derived in the same way as for the centroid jitter.

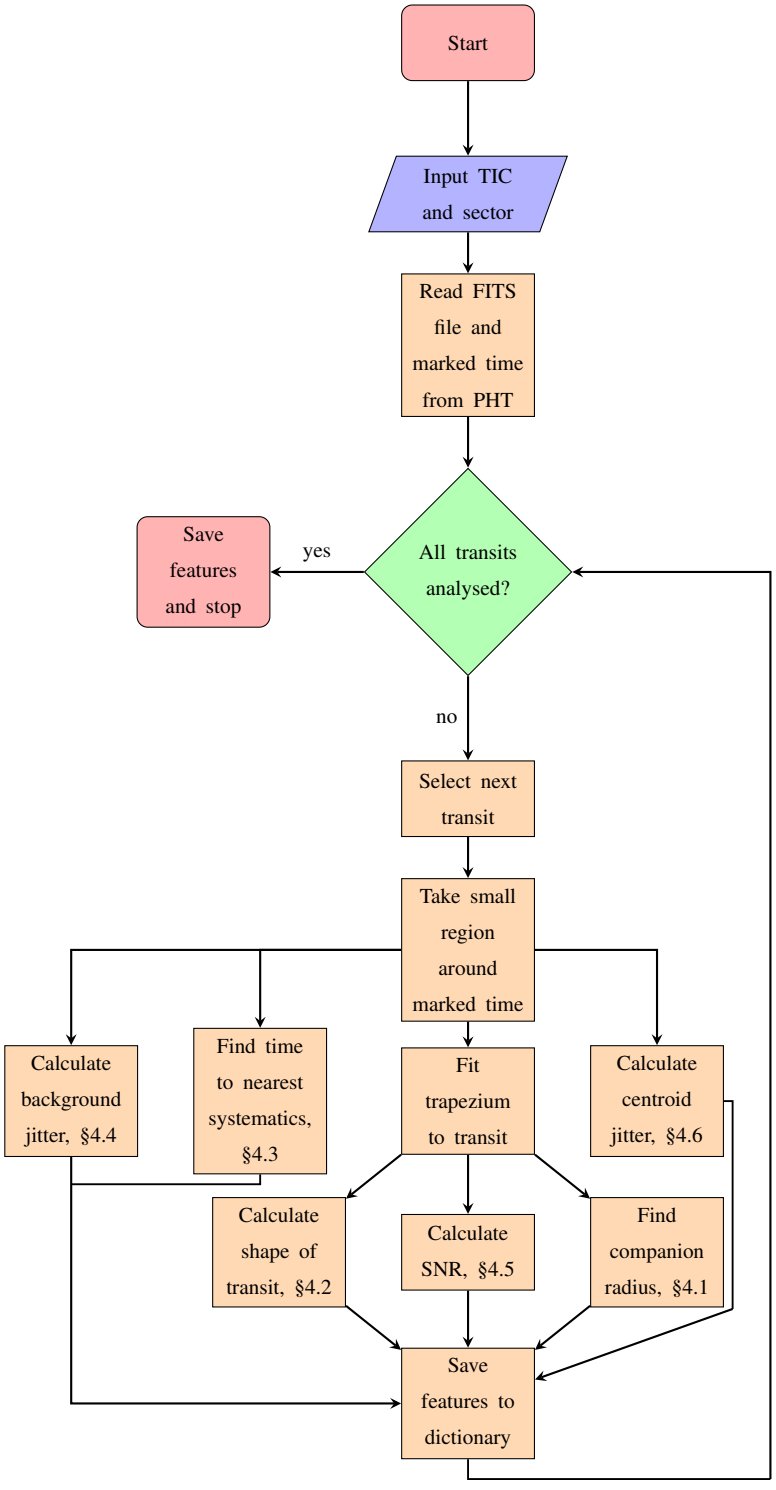

Figure 1: A schematic flowchart of the feature extraction algorithm. The above procedure is applied to each star in each observational sector separately. The analysis is run in Python 3.6 and takes approximately 10 seconds for each transit.

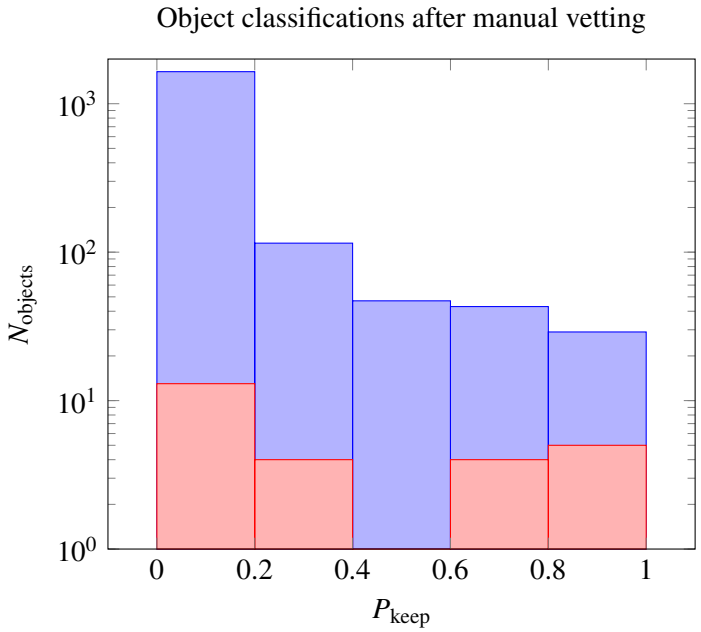

Figure 2: The Frequency distribution of all 1878 light curves (blue), or the subset of 26 TOIs(red), receiving a fraction P_{keep} of votes to keep at manual vetting stage. There is rarely a unanimous opinion of whether a marked transit is planetary in nature due to the fact that each light curve is examined by at least three people.

5 Results and Discussion

5.1 Training set

The training set for both the RF and PRF classifiers consists of 1878 light curves from sectors 12 to 15 which have been manually vetted. The light curves that have been manually vetted are seen by at least three independent vetters, therefore there is often disagreement on the designation of an individual light curve, as illustrated in Fig. 2.

Fig. 2 also shows that the vast majority of light curves are unlikely to contain planetary transits[11]. To illustrate why this poses an issue, consider a classifier which rejected all candidates without any feature analysis. For this data set, such a classifier would boast over 90% accuracy.

To attempt to balance the training set, the training set also includes features from all the known TOIs released by the *TESS* science team[12]. The list of TOIs contains the epoch (time of first recorded transit) and period of the companion orbit, as well as the sectors in which it has been observed. These ephemerides can be projected forwards to find the time of each transit in each observational sector; the light curves can be analysed as though they were detected through PHT. The transits are assumed to be planetary in nature and are treated as though all vetters had designated them as planets. Since most TOIs are detected by taking advantage of the periodic nature of transits, there is an observational bias towards short period orbits. As such, there will, on average, be more transits in a light curve identified from the list of TOIs than for a light curve identified by PHT. To avoid over-training on TOIs, only up to six randomly chosen transits for each TOI light curve are analysed. Appendix B illustrates the importance of a balanced training set for this classification problem.

It should be noted that the list of TOIs is not a list of planets and contains some false positives. In addition, fig. 2 illustrates that the majority of TOIs are rejected at manual vetting stage. However, for the purpose of this project, which aims to produce a method of automatically selecting the most promising candidates for detailed follow up rather than to simply emulate the current manual vetting process, it will suffice to hold these as ground truths.

The inclusion of the TOIs brings the total number of marked transits to 9704. This set is split in a 90:10 ratio to form a large training set and a smaller test set to verify the accuracy of the predictor.

5.2 Random Forest

The features extracted from the light curves are used as the training set for a Random Forest Classifier. This project uses the `sklearn` implementation (Pedregosa et al. 2012) which fails to work with data sets containing `NaN` values. To avoid completely discarding objects which may have only one missing feature, any missing features are predicted using the `Imputer` module built in to `sklearn`. The `Imputer` module predicts the value of features that were unable to be extracted (and given a `NaN` placeholder value). The imputer works iteratively: at each stage, a feature (the list of a given feature for all objects) column is designated as an output y and the others features are treated as inputs. The a regressor is used to learn a mapping $f : X \mapsto y$ for known y and predicts unkown y from this mapping. This is repeated for each feature. Since this will clearly not be as robust as actual feature extraction, light curves with more than one failed feature are rejected automatically.

The labels for each object were calculated according to three rules: a simple majority vote of the vetters, accepting any event which was marked as genuine by anyone, and rejecting all events that weren't unanimously marked as genuine. The results of these regimes, as well as the number of decision trees, on the overall accuracy of the classifier are summarised in Fig. 3.

[11] It is worth noting the light curves selected for manual vetting are the ones deemed most significant by the DBSCAN algorithm - the total set of light curves has an even more unbalanced distribution.

[12] https://tev.mit.edu/data/

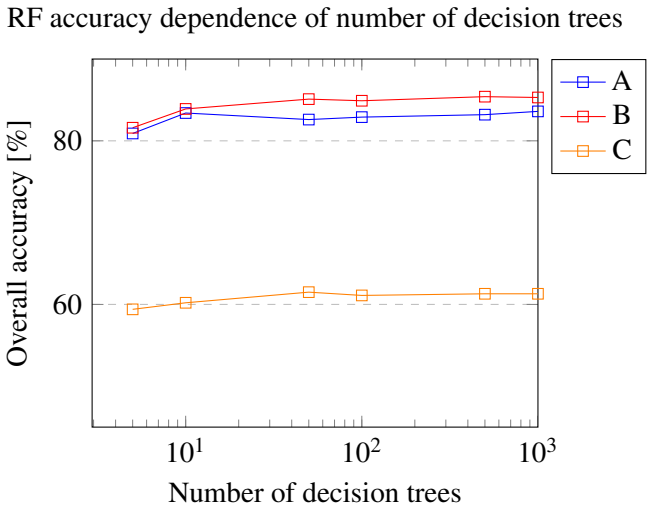

Figure 3: The Random Forest accuracy dependence on number of decision trees. The accuracy dependencies are shown for three regimes for deciding labels: accepting anything marked as genuine at least once (A), a majority vote (B), and rejecting everything any individual rejects (C).

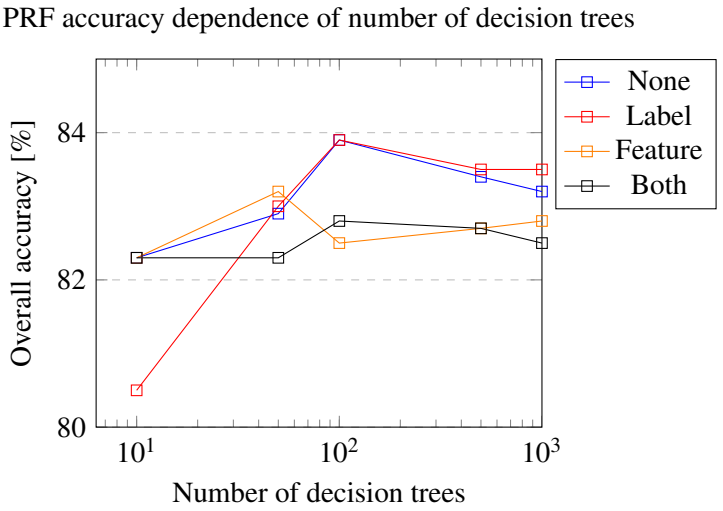

Figure 4: The Probabilistic Random Forest accuracy dependence on number of decision trees. The accuracy dependencies are shown for a data set without uncertainties, with just label or feature uncertainties, and for both label and feature uncertainties. Where no label uncertainties are used, labels are take as a majority vote.

The classifier performs similarly well if the labels are decided by majority vote or by accepting any object which is marked as genuine, boasting over 80% accuracy for sufficiently many decision trees.

By contrast, the accuracy of the classifier is significantly decreased if only the objects which are unanimously classified as genuine are accepted. In the latter case, the classifier has only a 61.3% accuracy for 1000 decision tree. It should come as no surprise that the classifier performs considerably worse for this case. Fig. 2 demonstrates that the vast majority of transit events have at least one vote against them so only considering the transits that are believed unanimously will vastly reduce the training set of 'true' transits. This will inevitably lead to a worse performance for any Machine Learning algorithm.

5.3 Probabilistic Random Forest

The Probabilistic Random Forest is publically available as a python package[13], designed to have a similar user experience to the `sklearn` package. Unlike the Random Forest, the PRF is capable of handling `NaN` values and no imputation is required. Furthermore, it can incorporate uncertainties in the features or labels of objects.

Fig. 4 highlights the effect of incorporating uncertainties to the classifier. It shows that the inclusion of label uncertainties offers a slight increase in classification performance whereas the feature uncertainties make the classifier less accurate. It is not quite clear why this should be the case but one possible reason is that the inclusion of uncertainties acts to homogenise the populations of the two classes, making definite classifications harder. Another effect could be that the uncertainties are not realistic. This could happen when the actual transit is too far away from the marked time and the fitting software fails to make a sensible prediction. An example of this is when there is a short period eclipsing binary, causing the clustering algorithm to combine the markings of individual transits and mark the out of transit region between events.

[13] https://github.com/ireis/PRF

5.4 Incorporation of multiple transit features

Up to this point, this work has aimed to treat all events individually to best preserve the ability of PHT to identify single transit events and mitigate the bias of traditional detection and classification pipelines towards shorter period signals. To illustrate the limitations of this approach, the classifiers were trained on the features extracted as in section 4 as well as the number of transits marked for that light curve. The results of this analysis are available in appendix C.

The inclusion of a multiple event metric, even one as simple as the number of events in a given light curve, have a dramatic increase in classification performance. The Random Forest gives a 94.7% classification accuracy and the Probabilistic Random Forest performs similarly well with 95.1% overall accuracy. This improvement should not be surprising when one considers that the majority of planet detection pipelines take advantage of the periodicity of signals.

6 Conclusions

This report has shown that the Probabilistic Random Forest classifier shows no distinct advantage over the classical Random Forest for the task of vetting light curves marked by Planet Hunters *TESS*. It has also shown that the inclusion of uncertainties for the Probabilistic Random Forest has no real improvement and in some cases can be worse than running the PRF with no uncertainties.

This project has also demonstrated that the treatment of transit events singly reduces the predicting power of a Random Forest classifier. This supports the work of McCauliff et al. (2015) who use the orbital solution of multiple transits to obtain a classifier with overall error as low as 5.85%. An interesting further project could be to expand this work to include more multiple event metrics (e.g. ephemeris matching or comparing odd-even transit depths), that were beyond the scope of this project.

As with all Machine Learning projects, this project would benefit from a larger training data set. Due to the nature of the classification process, this was impractical within the time frame of this project. A further improvement could be to look at the effect of including differential aperture flux as a feature for the Random Forest. By using the target pixel files (a small field of view around a target star), one can artificially create two light curves taken with different apertures. A light curve that has significantly different flux from the two apertures can signify that any variation is caused by another system at small angular separation (e.g. a blended eclipsing binary). This metric hasn't been used in this analysis for two reasons. Primarily, in future sectors it will no longer appear in the manual vetting stage, so an automatic classification that relies on this metric would not be a genuine representation of the vetting process. Secondly, the field of view must often be visually verified to ensure that the aperture chosen is sensible and doesn't, say, include a nearby but distinct star. Such a verification task is beyond the scope of this project but may warrant further investigation.

Acknowledgements

I would like to thank my supervisors; Suzanne Aigrain, Oscar Barrágan, and Nora Eisner, without whom none of the preceding work would have been possible.

References

Ansdell, M., Ioannou, Y., Osborn, H. P., et al. 2018, The Astrophysical Journal, 869, L7

Armstrong, D. J., Pollacco, D., & Santerne, A. 2016, Monthly Notices of the Royal Astronomical Society, 465, 2634–2642

Borucki, W. J., Koch, D., Basri, G., et al. 2010, Science, 327, 977

Breiman, L. 2001, Mach. Learn., 45, 5–32

Colon, K. 2019, TESS Science Support Center

Eisner, N. L., Barragán, O., Aigrain, S., et al. 2020, Planet Hunters TESS I: TOI 813, a subgiant hosting a transiting Saturn-sized planet on an 84-day orbit

Fischer, D. A., Schwamb, M. E., Schawinski, K., et al. 2011, Monthly Notices of the Royal Astronomical Society, 419, 2900–2911

Hippke, M., David, T. J., Mulders, G. D., & Heller, R. 2019, The Astronomical Journal, 158, 143

Howell, S. B., Sobeck, C., Haas, M., et al. 2014, Publications of the Astronomical Society of the Pacific, 126, 398

Mayor, M. & Queloz, D. 1995, Nature, 378, 703

McCauliff, S., Jenkins, J., Catanzarite, J., et al. 2015, The Astrophysical Journal, 806

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2012, Scikit-learn: Machine Learning in Python

Rauer, H., Catala, C., Aerts, C., et al. 2014, Experimental Astronomy, 38, 249–330

Reis, I., Baron, D., & Shahaf, S. 2018, The Astronomical Journal, 157, 16

Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2014, Journal of Astronomical Telescopes, Instruments, and Systems, 1, 014003

Schwamb, M. E., Lintott, C. J., Fischer, D. A., et al. 2012, Astrophysical Journal, 754, 1

Schwamb, M. E., Orosz, J. A., Carter, J. A., et al. 2013, The Astrophysical Journal, 768, 127

Shallue, C. J. & Vanderburg, A. 2018, The Astronomical Journal, 155, 94

Tenenbaum, P. & Jenkins, J. M. 2018, TESS Science Data Products Description Document, Tech. rep.

Thompson, S. E., Coughlin, J. L., Hoffman, K., et al. 2018, The Astrophysical Journal Supplement Series, 235, 38

Wang, J., Fischer, D. A., Barclay, T., et al. 2015, The Astrophysical Journal, 815, 127

Appendix

A PHT interface

Fig. 5 demonstrates the user interface of the PHT platform. Citizen Scientists are presented with a light curve and have the ability to zoom in to a section if necessary. Transits are selected by drawing a box around the transit events. Not drawing any boxes is taken to indicate that there are no transits in this light curve. After selecting transits, users have the option of proceeding to another light curve (via the 'Done button), or discussing this light curve in the 'Talk' forum (via 'Done Talk'). In the forum, users can directly bring particularly interesting targets to the attention of the research team.

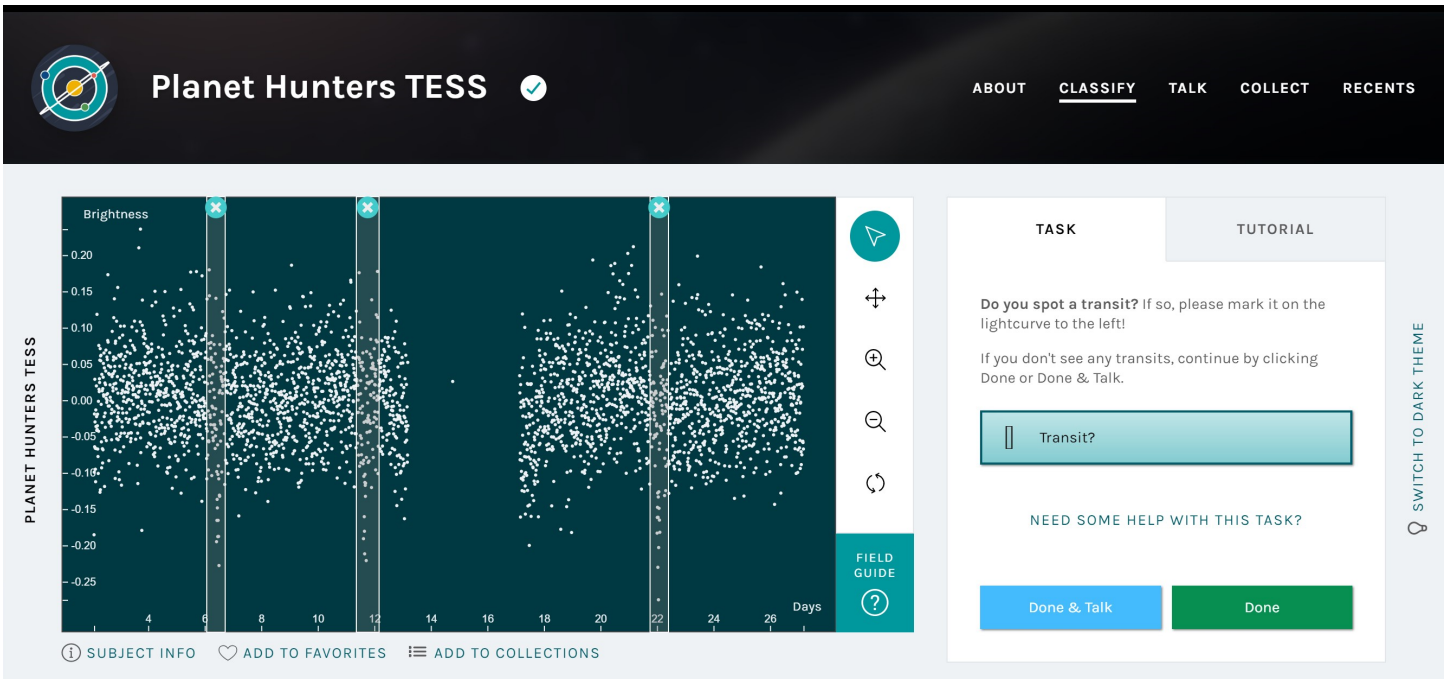

Figure 5: The PHT interface for an example light curve as seen by the Citizen Scientists. This particular light curve contains three injected transits (indicated by the blue boxes) to assess the Citizen Science approach to transit detection.

B The effect of excluding TOIs

Tables 1 and 2 show the effect of excluding TOIs from the training set. In both cases, the classifier designates all objects as false positives (i.e. predicts all objects as 'not planets'), regardless of features. Despite the fact that this classifier clearly has no predictive power, it reports a very high overall accuracy due to the imbalance in the test set. Therefore, for test sets with a large imbalance, one must be careful when quoting the overall accuracy as the metric for classifier success. Since the main work has used a more balanced training set, this distinction has not been necessary.

Table 1: Confusion matrix for classification of marked transits with a Probabilistic Random Forest on a training set with no TOIs. There is a 96.5% overall accuracy. Labels are chosen by a majority vote of vetters. The PRF is run without uncertainties and with 1000 decision trees.

n = 317	Predicted positive= 0	Predicted negative = 317
Actual positive = 11	TP = 0 (0.0%)	FN = 11 (100.0%)
Actual negative = 306	FP = 0 (0.0%)	TN = 306 (100.0%)

Table 2: Confusion matrix for classification of marked transits with a classical Random Forest on a training set with no TOIs. There is a 96.5% overall accuracy. Labels are chosen by a majority vote of vetters. The RF is run with 1000 decision trees.

n = 317	Predicted positive= 0	Predicted negative = 317
Actual positive = 11	TP = 0 (0.0%)	FN = 11 (100.0%)
Actual negative = 306	FP = 0 (0.0%)	TN = 306 (100.0%)

C The effect of multiple event features

Tables 3 and 4 show the effect of including the number of transits in a light curve as a feature for the RF and PRF algorithms. There is a significant improvement for both classifiers.

Table 3: Confusion matrix for classification of marked transits with a Random Forest. The number of transits in each light curve have been used as a feature. There is a 94.7% overall accuracy. Labels are chosen by a majority vote of vetters. The classifier is run with 1000 decision trees.

n = 971	Predicted True= 646	Predicted False = 325
Actual True = 661	TP = 628 (95.01%)	FN = 33 (4.99%)
Actual False = 310	FP = 18 (5.81%)	TN = 292 (94.19%)

Table 4: Confusion matrix for classification of marked transits with a Probabilistic Random Forest. The number of transits in each light curve have been used as a feature. There is a 95.1% overall accuracy. Labels are chosen by a majority vote of vetters. The classifier is run without uncertainties and with 500 decision trees.

n = 791	Predicted True= 468	Predicted False = 323
Actual True = 477	TP = 453 (94.97%)	FN = 24 (5.03%)
Actual False = 314	FP = 15 (4.78%)	TN = 299 (95.22%)