MNRAS **000**, 1–11 (2015) Preprint 19 September 2017 Compiled using MNRAS LATEX style file v3.0

Machine Learning for Exoplanet Detection

J. Gamper,[1]★ D. J. Armstrong,[2]† T. Damoulas[3]†
[1]*Mathematics of Systems, University of Warwick, Gibbert Hill Road, Coventry, CV4 7AL, UK*
[2]*Department of Physics, University of Warwick, Gibbert Hill Road, Coventry, CV4 7AL, UK*
[3]*Department of Computer Science, University of Warwick, Gibbert Hill Road, Coventry, CV4 7AL, UK*

19 September 2017

ABSTRACT
When searching for new planets through transit detection in NASA's Kepler satellite data, a significant portion of time is spent on validation of the detected signal. For example, light curve fitting of the possible false positive scenarios, and follow-up observations using other methodologies. Using pre-trained models would allow researchers to quickly validate planets and investigate only uncertain signals, particularly in future, data abundant, missions such as PLATO. We optimize and evaluate the performance of multiple machine learning techniques in classifying planet candidates into false positive and confirmed planets, using publicly available Kepler light curve data. The data is transformed into real-numbered attributes through Kepler science pipeline and additionally computed attributes, such as ephemeris correlation and statistic derived from self-organized maps. First, we test the classification accuracy and provide interpretation to the results obtained. Second, we evaluate the quality of probabilities obtained from machine learning models against those acquired from MCMC based planet validation method - **vespa**. Identifying a significant disagreement between probabilities for yet not dispositioned planets in Kepler data, we test the latter for novelty. To optimize the interpretability of machine learning for planet validation we test sparse Gaussian process classification, which has the advantage of scalability and posterior variance in prediction. High posterior probability variance was not achieved for false positive instances where machine learning predicts high false positive probability contrary to **vespa**, likely because of some of the false positive scenarios appearing in disposition data not accounted for by the MCMC based model. The converse appears to be the case for not dispositioned data, where **vespa** provides confident probabilities and machine learning methods are less certain. Signaling a possible difference in false positive cases composition in labeled and unlabeled data. Most of machine learning models capture 99% of false positives and 74% to 92% of confirmed planets, with Gaussian process based model identifying misclassified planets as high uncertainty predictions, pointing out instances where researchers input is needed.

Key words: planets and satellites: detection; planets and satellites: general; methods: data analysis; methods: statistical; methods: machine learning

1 INTRODUCTION

Imagine trying to find a firefly in front of the brightest man-made spotlight at a distance of hundreds of miles away from the observer. This analogy serves to describe a challenge for astronomers across the world, and engineers designing detectors able to work at such precision. Light reflected by the planet onto the observer is 10^{-9} times fainter than the brightness of a hosting star (Mason 2008). The planet is essentially lost within the glare of the star, making the search for earth-like planet by direct observation inefficient. Hence, the existence of a planet is inferred indirectly by observing the star itself.

As an example of an indirect detection methods is the Doppler effect technique, also known in the scientific literature as the radial-velocity method, proposed by Struve (1952). The Doppler effect method relies on radial-velocity measurements, in particular on the doppler shifts caused by the star orbiting a common center of mass between itself and the planet, producing a wobble like effect (Karttunen et al. 2007). The precision currently achieved is able to register

★ E-mail: j.gamper@warwick.ac.uk
† D. J. Armstrong and T. Damoulas co-supervised the MSc project, as part of the CDT Mathematics of Systems curriculum at Warwick Mathematics Institute

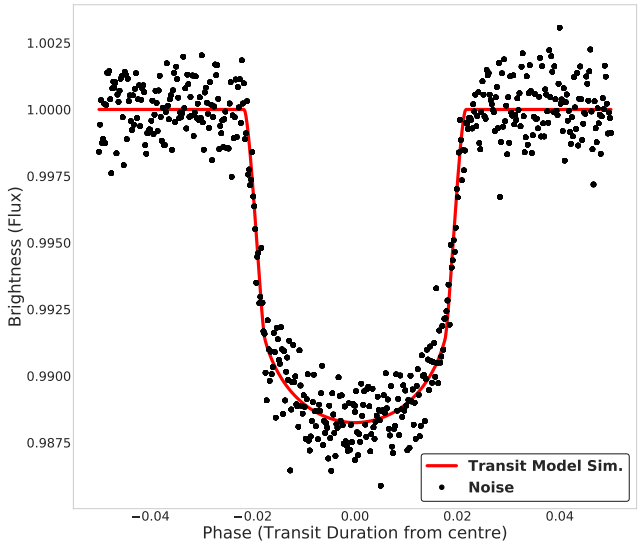

Figure 1. A simulation of a phase folded transit of a planet with period of 4 days. Simulated using the model of Mandel & Agol (2002), using the publicly available code of Parviainen (2015).

a wobble effect of approximately $1\frac{m}{sec}$, naturally leading to discoveries of gas giants at most (Borucki et al. 2003). The detection of an Earth-size planet would require an instrument able to register a wobble effect at the scale of $\frac{cm}{sec}$. As an alternative, transit photometry - an indirect planet detection method where brightness of a star is measured precisely as a function of time, could register a dim in the flux (brightness) hopefully produced by an earth-sized planet's shadow swiping across the telescope's detector. An earth-like planet produces one ten thousandth of change in brightness of a star, compared to Jupiter like planets producing a change of around one percent. Besides the technological challenge of producing such precision, a probability of an earth like analog orbiting a star at a convenient angle of sight is just .5 percent. To find just one planet we would have to observe 200 different stars. Therefore, under the observation field of the Kepler mission there are about four and a half million stars, of which about hundred and fifty thousand were hand picked for transit-photometry. Those stars which are suspected to harbor a planet are subsequently designated as Kepler Object of Interest (KOI). Working with such sparse signals requires robust statistical methods to process the data coming from the Kepler satellite and validate each selected candidate as **confirmed** planet or a **false positive** (FP). In fact, Brown (2003) identified 12 different combinations of giant planets and eclipsing or transiting star systems that may produce a signal similar to that of a planet transiting a star. Some of these scenarios may also contribute to true positive detected signals by distorting the estimated size of the planet (Bryson et al. 2013). From the light curve time series obtained by Kepler spacecraft, some of the characteristics of a planet can be inferred based on Kepler's laws of planetary motion. For a more in-depth introduction to planetary motion refer to Sackett (1999), and for a more realistic introduction refer to Winn (2014); Murray & Correia (2010). For an example of analytic and modelling work on transit light curves refer to Mandel & Agol (2002). For an in-depth report on the Kepler missions achievements and contributions to astronomical community, refer to Batalha (2014).

In this work we test the robustness of machine learning classification methods in identifying planets and false positives, and compare the results to existing, more traditional, validation methods based on Markov Chain Monte Carlo simulations. Automation provided by verified machine learning methods would allow speed ups on validation process and point to uncertain instances where researcher's input is needed for validation. Proof of concept of machine learning methods for planet validation would be particularly important for future missions such as PLATO [1] and NASA TESS, where the amount of data will be in an order of magnitude larger. Below, we provide a brief outline of our work. In section 1.1 we review the relevant parts of the Kepler Science pipeline that generate inputs for our algorithms. Section 1.2 reviews the existing planet validation methods and emphasize the methods we are comparing against. Section 1.3 provides a brief overview of machine learning methods applied to Kepler data. In section 2 we describe the methods, data and the results of our work, and conclude in section 4.

1.1 Kepler science pipeline review

The Kepler Mission Science Operation Center (SOC) at NASA Ames Research Center is primarily responsible for processing the data obtained from Kepler single instrument spacecraft, for example managing the target stars, and processing the raw photometric data downlinked from the spacecraft each month via Deep Space Network. SOC searches each flux time series for signatures of the transiting planets, performs fitting for physical parameters of the planet, and obtains statistical significance scores, some of which contribute as inputs to the machine learning methods tested in this work. The goal of the SOC mission is to characterize the frequency of exoplanets with respect to diameter, orbital period and host star as reviewed in Borucki et al. (2010) and Koch et al. (2010). The SOC science pipeline is divided into several components. The most relevant components producing statistics used as model inputs in this paper are reviewed in this section. For a more elaborate review of the SOC pipeline refer to Jenkins et al. (2010).

SOC receives unprocessed data from Kepler spacecraft. Most of it is science data, however the engineering related data is included as well. For example, temperature of the focal plane and readout electronics - as it may reflect the quality of the science data and assist during the post processing stages on the ground. Subsequently the calibration module produces the correct pixel data and estimates the background pixels. These are processed by the Photometric Analysis (PA) module to fit and remove the sky background, and measure the target star centroid location. Target star centroid locations play a significant role in the downstream analysis, and one of the resulting features used in our data heavily relies on the centroid analysis. In particular, it allows us to estimate if the actual transit signal is produced by the target star. I.e. PA produces difference images which

[1] Results obtained in this work will be presented at *The PLATO Mission Conference 2007: Exoplanetary systems in the PLATO era* on 05-07/09/2017.

are compared to the out of transit centroid, and if there is a significant offset a background eclipsing binary may have produced the transit-like signal.

The final step in producing the light curves happens in Pre-Search Data Conditioning where signatures in the light curves correlated to known instrumental artifacts are removed. Examples of the instrumental artifact are: pointing drift of the spacecraft, thermal effects or focus changes. After this stage, the data produced includes raw and calibrated pixel data, raw and systematic error-corrected flux time series, centroid measurements and associated uncertainties for each target star. Further, in the Transiting Planet Search (TPS) module a wavelet-based adaptive matched filter of Jenkins et al. (2002) is applied to identify transit like features. Light curves with transit like features whose combined folded transit detection statistic exceeds 7.1σ for some trial period and epoch are designated as threshold crossing events (TCE), where TCE is a time series sequence with significant planet transit-like features in the light curve of the target star. Selected TCEs are passed into the Data Validation (DV) module, which performs statistical test to evaluate the confidence in detection in order to reject **false positives** (FP) by background eclipsing binaries, and to extract physical parameters of each system along with uncertainties and covariances for each TCE. Most of the DV-produced values are inputs into our preprocessing steps discussed below. After the planetary signature has been fitted, it is removed from the light curve and the residuals are subject to search for additional transiting planets. The process repeats until no further TCEs are identified.

After the DV module the obtained TCEs and data are passed to the Science team for subsequent validation of the planets as **confirmed**, unless sufficient evidence is found for one of the FP scenarios after the careful consideration of the light curve of the TCE. The light curve for a TCE is constructed by phase folding the data obtained from the DV module given the parameters received. In Figure 1 we simulate a phase folded light curve for a planet with the period of 4 days, and planet to star radius of 0.1, with normally distributed noise around the simulated light curve. This is to serve as an example of the data that the validation models discussed below in section 1.2 are fit to.

1.2 Kepler planet validation review

In comparison with ground based exoplanet surveys, the Kepler mission is different in terms of volume of candidates to be validated and in limits of traditional validation techniques such as radial velocity discussed above, particularly for small planets and/or faint stars. This has led to the adoption of *probabilistic validation* methods. The Kepler mission team introduced a **blender** procedure, where a team of researchers uses a combination of follow up observations and light curve modeling to rule out the parameter spaces corresponding to possible FP scenarios, allowing for the evaluation of relative likelihood of confirmed planet to that FP hypotheses (Torres et al. 2010). Although the **blender** method partially solves the problem of validation, and in fact has been quite successful, as in Fressin et al. (2011), it is demanding in terms of researchers labor and repetitively used CPU time which limits the number of the FP scenarios explored and the number of candidates validated.

As reiterated by Morton (2012), the primary goal of the Kepler mission is to obtain population statistics, this has inspired the author to introduce an automatic method for FP probability estimation for any given Kepler candidate using only the features available from Kepler spacecraft, such as period, depth, duration, and shape of the signal, as well as, the features of the star, along with the informed prior assumptions of the stellar properties Morton & Johnson (2011). In a follow-up paper Morton (2012) demonstrated that his automatic probabilistic validation method can reliably identify FP signals, and has been used to validate KOI-961 planets (Muirhead et al. 2012). Many candidates that have passed the vetting in the Kepler science pipeline based on transit-photometry alone or methods such as **blender** have the FP probability of around 5%. As of current dispositioned Kepler data, the average FP probability according to the validation model for confirmed planets is 3%.

The validating procedure introduced by Morton & Johnson (2011), commonly abbreviated as **vespa**, can be summarized as follows. For any candidate TCE to be validated, we need to demonstrate that the probability of the FP scenario is small enough to be considered negligible (Morton 2012). In particular, for any planet a FP probability is defined as:

$$P(FP) = 1 - P(planet|signal), \tag{1}$$

where

$$P(planet|signal) = \frac{\mathcal{L}_{TP}\pi_{TP}}{\mathcal{L}_{TP}\pi_{TP} + \mathcal{L}_{FP}\pi_{FP}}. \tag{2}$$

Where π in the above equation stands for a prior of a certain scenario labeled as TP (true positive) or FP (false positive) standing for all false positive scenarios, and $\mathcal{L}$ represents the likelihood for a given scenario $P(D|H_{FP})$, with D standing for observed data, and H_{FP} for false positive hypothesis. For a thorough introduction on Bayesian model selection and hypothesis testing in physical sciences refer to Gregory P. (2005). To compute the above mentioned probabilities, a representative population is simulated for each scenario using the period of the transit signal. From the simulated representative population a prior π is obtained for each scenario. At this point, an additional information might be added into the prior - for example information from follow-up observations. The model likelihood $\mathcal{L}$ is computed using a Markov Chain Monte Carlo routine for each scenario given the candidate data. Lastly, all of the quantities are combined to compute $P(FP)$. If the value is less than 1% the planet is considered to be validated.

Diaz et al. (2014) introduced a more rigorous validation method **pastis**. However, the complexity introduced with **pastis** fosters substantially slower computation - a typical **pastis** run would take from several hours to tens of hours, where **vespa** would take from several minutes to an hour. The primary difference between **vespa** and **pastis** is that **vespa** reduces the information about the light curve down to three parameters, such as duration, depth, and ingress and egress phases. This is done by fitting only one non-physically representative trapezoidal model to the light curve, independently of the FP scenario considered. According to Diaz et al. (2014) this is sufficient only for current missions like Kepler. Future space mission such as PLATO would require

us to take a full advantage of the light curve data, and fit an appropriate light curve model given the FP scenario considered. As a result, **pastis** attempts to obtain the Bayesian odds ratio between planet and all FP hypothesis, using all the available information, including the follow-up observations and models specifically chosen given the FP scenarios considered:

$$P(H_i|D,\pi) = \frac{P(H_i|\pi)P(D|H_i,\pi)}{P(D|\pi)}, \tag{3}$$

After obtaining the posterior probabilities for each hypothesis, the goal is to compute the odds ratio (2).

$$O_{ij} = \frac{P(H_i|D,\pi)}{P(H_j|D,\pi)} = \frac{P(H_i|\pi)P(D|H_i,\pi)}{P(H_j|\pi)P(D|H_j,\pi)} \tag{4}$$

where the odds ratio is expressed in terms of Bayes factor (ratio of global likelihoods) and prior odds. Just as for the computation of the likelihoods for VESPA, computing the global likelihood (evidence) requires an integration over the parameter space for the corresponding hypothesis H_j:

$$P(D|H_i,\pi) = \int P(D|\theta_i,H_i,\pi)P(\theta_i|\triangle)d\theta_i, \tag{5}$$

with the threshold value for the odds ratio specified a priori.

1.3 ML-based scientific applications review

All three of the validation methods above require a model likelihood $\mathcal{L}$ to be computed at some stage of the validation process, either with a handcrafted **blender** approach, or simulation based approach in **vespa**, or the Bayesian extrasolar planet validation method **pastis**. Unfortunately, these likelihoods require an integration over a parameter space of several dimensions leading to time consuming MCMC methods. Therefore, scientific community has turned to testing the applicability of machine learning tools in their field. As an example, hypothesis testing as likelihood-free inference expressed as density ratio can be performed by building a classifier as shown by Neal (2008), and applied to high energy physics by Kyle et al. (2015), where the abundance of data coming from the LHC makes a MCMC approach impractical. Equally, machine learning methods such as dimensionality reduction have been applied to transit light curves: Matijevic et al. (2012) applied Local Linear Embedding model to characterize the light curves produced by eclipsing binaries; Thompson et al. (2015) used Local Preserving Projections to map the light curves to lower dimensions to differentiate transit and non-transit like signals; and Armstrong et al. (2016) has obtained disposition statistic based on proximity of the transit light curves in a space projected by Self-Organizing Maps, which we use as feature input in this work. McCauliff et al. (2014) used random forests to learn the mapping from features derived from Kepler data to labels of candidate planet, astrophysical false positive, and non-transiting phenomena, which is one of the first attempts to classify the nature of the signal, but not to validate the planet. We reconstruct the feature selection process of the latter, however we do not replicate as many features and achieve comparable result on binary classification directly into Confirmed planets or False Positive, as discussed in the sections below.

2 DATA PREPROCESSING

2.1 Introduction

Following the procedures in McCauliff et al. (2014) we use the TCE catalog[2] from the 17 quarters of Kepler data. Which in total contains 34,024 TCEs. These TCEs are matched with the cumulative KOI catalog[3] of known planets from the NASA Exoplanet Archive, which is the union of many KOI catalogs. For a more detailed description of the KOI catalogs refer to Borucki et al. (2011), Batalha (2014), Burke et al. (2014). Matching TCEs with known planets in KOI catalog produced a training and testing dataset of 4049 planets, of which 2238 were **confirmed**, and 1810 were **false positive**s (FP). The labels for the known planets were determined using one or a combination of the methods described in section 1.2. For some of the TCEs several features might be missing, which may be due to missing information in stellar catalogs, or due to DV module in Kepler Science pipeline fitting methods not converging or reaching a time out. Following McCauliff et al. (2014) we use class conditional means to substitute for missing values. Any features which had number of missing values greater than 60% have been removed, leaving a total of 139 features out of the original 149 obtained from the TCE and KOI tables.

2.2 Maximum Ephemeris Correlation

We extended the feature space by creating a maximum ephemeris correlation feature following McCauliff et al. (2014). Using the orbital period p and transit epoch t_0, we are able to reproduce the transits and compute the Pearson's correlation coefficient between a given TCE and other TCE's that are hosted on the same star. Namely, for each TCE we compute a binary transit indicator array z_k, where 1 stands for in transit, and 0 for out of transit, with an observation window of 1 minute. The ephemeris correlation ρ_{match}, described in Equation 6 is computed between every TCE's on the same star. If there is only one TCE on the same star then this value is zero, otherwise the maximum computed value is the statistic recorded for that TCE.

$$\rho_{match} = \frac{z_k}{\sqrt{z_k}} \cdot \frac{z_j}{\sqrt{z_j}} \tag{6}$$

Highly correlated TCEs on the same star my indicate a signal detected by Kepler TPS module whose origin may be of secondary eclipse, and should be classified as FP. Maximum ephemeris correlation turned out to be one of the most significant features for planet classification.

[2] See https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=tce for a full TCE table.
[3] See https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative for a full KOI catalog table.

2.3 SOM

To the existing 150 features, we add an additional feature produced by Armstrong et al. (2016). The feature was derived from the results of applying Self Organising Map (SOM), an *unsupervised* machine learning algorithm that finds clusters in the dataset without any knowledge of labels. SOM is an artificial neural network, which learns the mapping from the input space into a latent two dimensional space. However, instead of using error-correction based learning such as traditional back-propagation based on stochastic gradient descent (Schmidhuber 2015), it uses competitive learning, where each individual data point competes to be a member of a certain cluster determined by a distance metric. Additionally, SOM preserves the topological structure of the high dimensional data when mapping to a lower dimensional space. For a more detailed overview of the SOM method refer to Kohonen (1990).

In the case of Kepler data, a lower dimensional representation of the Kepler transit light curves is obtained using SOM, and disposition statistic θ_1 is computed. For each low dimensional point (x_i, y_i) on a SOM a proportion of confirmed planets and false positives is computed, as well as the total number of known dispositions. These known dispositions will serve as weighting for the statistic, as the already dispositioned points should have more classification power. The weighting W and proportions $\alpha_{planet}(x, y)$ are calculated as follows:

$$W(x, y) = \sum_o (x_o = x, y_o = y), \tag{7}$$

and

$$\alpha_{planet}(x, y) = \frac{\sum_i (x_o = x, y_o = y)}{W(x, y)}, \tag{8}$$

where o is an index representing already dispositioned object as **false positive** or **confirmed**. The statistic used as a feature is then obtained by

$$\theta_1 = \frac{\sum_i (\alpha_{planet}(x_i, y_i) W(x_i, y_i)}{\sum_i (W(x_i, y_i))}. \tag{9}$$

2.4 Feature Importance and Correlation

After the preprocessing step, we continue with the reduction of the data feature space by estimating the importance of each individual feature. For each decision tree within a fitted to the data random forest, we permute the attribute within the out-of-bag data and recompute the error rate. Refer to Friedman et al. (2001) for a thorough description of decision tree and random forest models. The importance of the feature is a mean increase in error over all the estimators in the random forest using the out-of-bag data with the permuted attribute. Further, we compute a class-conditional importance by permuting only the rows corresponding to selected label, and for comparison use only the maximum estimated importance of the two classes. We proceed by estimating the importance of an insignificant attribute by fitting one hundred random forest models, each with a new random attribute. This exact procedure has already been used by McCauliff et al. (2014), where they found the cutoff point to be 6σ of estimated random attribute importance, which is what we used.

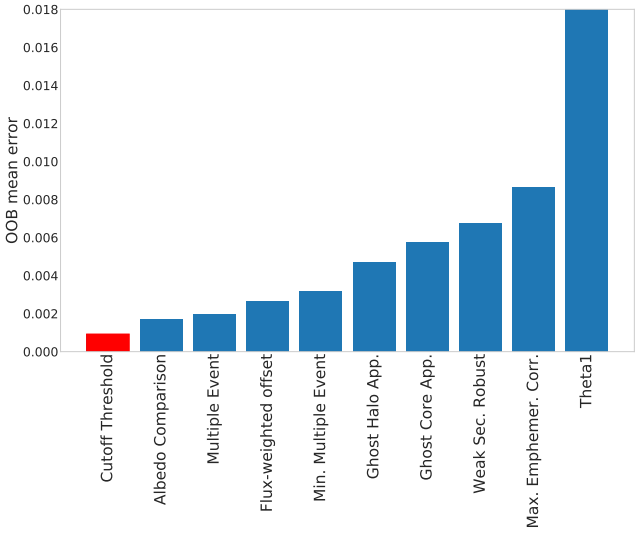

Figure 2. Out-of-bag feature importances estimated using random permutation for final, selected features after passing the importance, correlation screening and vetting. Cutoff threshold of a 6σ of estimated random attribute importance is included as well.

An alternative feature selection method we could have considered is impurity gain, although both the later and random permutations have some pitfalls. Strobl et al. (2015) has shown that the impurity gain tends to be in favor of variables with more varying values. While random permutation tends to overestimate the variable importance of highly correlated variables. Therefore, we compute Pearson's correlation coefficient for each pair of attributes. If it is larger than the threshold then the attribute with the least class-conditional importance is dropped. The correlation threshold was estimated by training a random forest using a grid of threshold values, and picking the one producing the least out-of-bag error.

2.5 Most Important Attributes

Out of the 33 features that have passed the importance and correlation filtering, we proceeded further by removing some of the attributes, as some of them would be assumed to not have any predictive value, and could have impacted the classification contribution due to errors within the Kepler KOI or TCE tables. The importances of the selected features are presented in Figure 2. The most important features Theta 1, and maximum ephemeris correlation have been described above. Below, we discuss some of the other important attributes, ordered according to their importance.

- Weak Secondary Robust Statistic - this statistic is computed within the DV module of the Kepler pipeline. It measures the significance of the transit signal after removing the outliers observations. The lower the value of the statistic the less likely TCE has to be an astrophysical FP.
- Ghost Core Aperture Statistic - correlation measure between the transit model used within the DV module of SOC, and the average flux per pixel in the core aperture. The core

Table 1. Top-down, the best performing models with their parameters, and the total number parameter combinations considered, see footnote 4 for a table of parameters considered. Latent and Quadratic Discriminant, Ridge and Logistic regression classifiers were not included in the table as these were tested with default parameters of scikit-learn package. All models had performance above 90% accuracy on the training set.

Classifier	Best Param. Selected	No. of combinations
Random Forest	100 DTs, Max Depth	4
Extra Trees	100 DTs, Max Depth	16
Decision Tree	Depth 10	3
AdaBoost	SAMME, DTs	3
K-NN	L1 dist., dist. weight.	24
SVM	Linear,10	6
Neural Network	(5,10),adam,reg:0.0001	327

aperture is the optimal light collection region on the objective lens of the space craft for that particular target star.

- Ghost Halo Aperture Statistic - measures the correlation between the transit model and the annulus surrounding the optimal aperture. Relationship between core and halo aperture statistics measures how likely is that the transit signature was produced by an optical illusion.
- Flux-Weighted Offset Significance - measures the significance of the difference between in-transit and out-of-transit flux weighted images of the target star. If the value is high, it is more likely that the transit detected is associated with the host star and is unlikely to be caused by other phenomena.
- Multiple Event Statistic (MES) - this statistic measures the quality of the transit event as determined by TPS module of SOC. For a target star, if multiple TCEs are found only the maximum MES is recorded for that star. However the ratio of minimum MES to maximum MES is recorded as well. When the ratio is close to 1, the threshold crossing events found below median flux are of similar significance as those found above median flux of the star, signifying concerning stellar variability or presence of instrumental noise in the light curve.
- Albedo Comparison Statistic - measures the difference between geometric albedo (the ratio of actual brightness of the object under consideration to that of a uniform disk) of the secondary event on a target start and 1. If the MES value is significant for the given TCE, and albedo comparison statistic is high, the TCE is likely to be a FP.

3 METHODS AND RESULTS

As reviewed in section 1.3, there have been individual cases where machine learning algorithms have been applied to data of astronomical origin to help answer specific questions. In this paper we parametrize, validate, and test a broad range of machine learning algorithms: Support Vector Machine; Logistic Regression; Ridge Logistic Regression; Decision Tree; Random Forest; Extra Trees; K-Nearest Neighbor; Multilayer perceptron (Neural Network); Linear Discriminant Analysis (LDA); Quadratic Discriminant Analysis (QDA); and ensemble of those using Boosting. We also test a sparse Gaussian Process classifier for estimating posterior uncertainty in predicted probabilities (James et al. 2015). In order to evaluate their capability to classify TCE signals into confirmed planets, and FP, and to compare if machine learning algorithms can obtain similar probabilities as MCMC based physical model fitting methods, specifically **vespa**. For all models except Gaussian Process classification, we used the scikit-learn Python software (Pedregosa et al. 2011). The former was tested using GPflow Python software (Matthews et al. 2016). Refer to Friedman et al. (2001) for an overview of the tested models, and to Rogers & Girolami (2011) for an introduction into Gaussian processes for classification and regression.

3.1 Classification

Using the 4049 data points, with 9 features we first exhaustively search for the best parameters for each model, by splitting the labeled data into 60% of data for training with the remaining data left for validation. The training is then split into 10 folds, and each possible parameter combination for each model is fit and cross-validated on these 10 folds, in the end providing an accuracy distribution for every possible parameter combination for a given model. In total, 421 model/model-specific-parameter combinations[4] were cross-validated. In our case this did not pose a significant computational overhead, and an exhaustive approach was also beneficial to evaluate the variance within the performance. See Table 1 for models ordered according to their performance, their parameters, and the number of parameter combinations evaluated - all best parameter options scored with accuracy above 90% on a training set, with Random Forest scoring the highest with mean of 96.8% and standard deviation of 0.007.

The models were compared based on five metrics on a test set. Precision S_P, recall S_R, and brier score S_B, defined in equations 10, 12, 13 respectively, where T_P stands for true positive (i.e. false positive signal predicted as false positive), F_P the number of confirmed planets classified as false positive, F_n the number of false positives signals classified as planets, and f_t and o_t stand for the predicted probability and actual outcome respectively. The precision is the fraction of data predicted as FP:

$$S_P = \frac{T_p}{T_p + F_p}. \tag{10}$$

$$S_{sp} = \frac{T_n}{T_n + F_p}. \tag{11}$$

$$S_R = \frac{T_p}{T_p + F_n} \tag{12}$$

Recall, also known as *sensitivity* describes how good the model is at detecting false positive signals within the Kepler dataset, i.e. the proportion of the false positive signals that were classified as false positive.

$$S_B = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{13}$$

[4] See https://github.com/jgamper/MScProject-Kepler-ML for a full training results and parameters considered csv file.

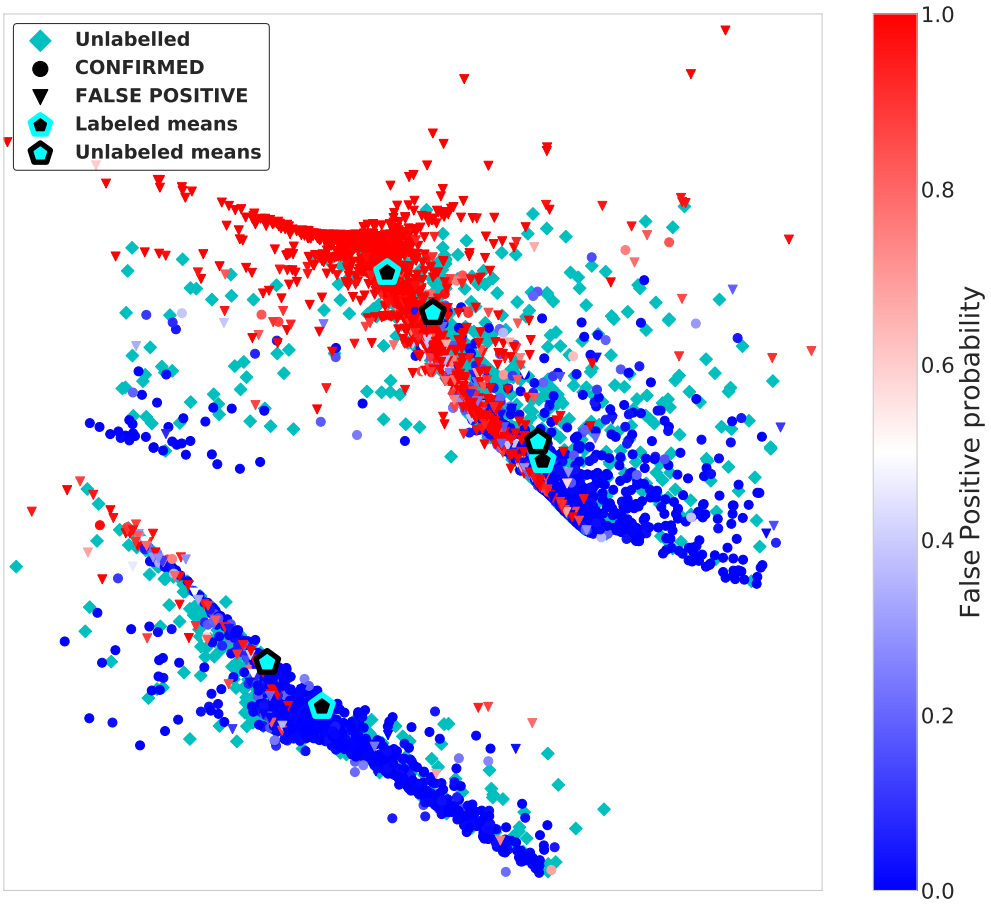

Figure 3. Kepler labeled and unlabeled data visualized using multi-dimensional scaling with FP probabilities extracted from random forest. Marker shape represents the true label: down triangle for false positive, circle for confirmed planet, and diamond for unlabeled data points. Cyan outlined and black outlined pentagons represent labeled and unlabeled data cluster means respectively.

The Brier score measures the correctness of probabilistic predictions of each model. The lower the score, the more accurate are the probability predictions of the classifiers.

Further, the metrics of recall and specificity are combined into an evaluation method called receiver operator characteristic (ROC) curves. Many classification algorithms provide a real valued probability output that has to be thresholded to give a classification. Therefore, for each classifier we vary the probability threshold and compute its precision and recall values. Resulting curves on the recall and precision axis would ideally reach to hit the top left corner, where both false positive prediction rate and true positive prediction rate is high, we do not plot the ROC curves as these are close to indistinguishable for our models, and only compute and discuss area under the curve (AUC) scores. The AUC results as well as other metrics can be found in Table 2.

Decision Tree based ensembles perform the best on the test set, as well as on the training set. Other models are very close in performance to random forest, with varying abilities to identify FP signal in the data judging by the recall. The top performers of DT ensembles are followed by relatively simpler classifiers, such as basic logistic regression, LDA, SVM and K-NN. In particular, it is interesting to note that in cases of discriminant analysis and support vector machines, linear variations performed better than QDA, or other complex kernel functions. The contrast is particularly, significant in the case of LDA and QDA, where QDA significantly under-performs in terms of correctly classifying the false positive signals. As reviewed in Bishop (2006), LDA assumes that the data is approximately Gaussian, and that the covariances of two classes are equal. According to Friedman et al. (2001), the discrepancy between QDA and LDA performance is an indicator of not necessarily the truth of the underlying model assumptions, but rather that the data

Table 2. Test set performance of the top classifiers using the parameters from Table 1 and optimized on the training set.

Classifier	AUC	Precision	Recall	Brier
Random Forest	0.99	0.96	0.94	0.03
Extra Trees	0.99	0.97	0.93	0.04
Logistic Regression	0.97	0.94	0.90	0.06
LDA	0.97	0.92	0.87	0.07
Ridge Classifier	0.97	0.92	0.87	0.10
SVM	0.97	0.92	0.90	0.17
K-NN	0.96	0.98	0.89	0.06
MLP	0.96	0.92	0.89	0.07
QDA	0.95	0.96	0.57	0.20
Ada-Boost (DT)	0.94	0.93	0.93	0.10
Decision Tree	0.92	0.95	0.93	0.06

can only support simple decision boundaries such as linear, and estimates provided via the Gaussian models are more stable. Essentially, we have a bias variance trade-off case - in our 9 dimensional space, the data can be linearly separated, with a few points appearing within the opposite cluster and scattered outside. The latter phenomena might be due to several reasons or a combination of such - low SNR ratio of data obtained by Kepler, and therefore the quality of the features that we have to separate these classes.

Decision Tree ensembles such as random forests would be expected to perform better than regular linear classifiers in the case where the data clusters well with a few outliers in some high dimension by randomly partitioning the space, while other more complex smooth models will perform worse trying to fit an overly complex smooth function. For example, the slacking performance of neural networks on this dataset. To explore this high-dimensional space, we tested several dimensionality reduction methods, finding that multidimensional scaling (MDS) captures the data best and provides easy interpretation. For MDS, we computed the Euclidean distance matrix between every instance, and use stochastic gradient descent to minimize the stress defined as:

$$\mathcal{S} = \frac{\sum_{i,j}(d_{i,j} - \sigma_{i,j})^2}{\sum_{i,j}\sigma_{i,j}^2}, \tag{14}$$

where $d_{i,j}$ is distance in the projected space, and $\sigma_{i,j}$ is distance in the original space. Using MDS, we are able to preserve the distances in the 9 dimensional space in two dimensional space. In Figure 3 we plot the labeled and unlabeled dataset together, along with the means found with the K-Means algorithm over 10 different initializations. We have tested the ability of K-Means to find cluster means as we use those as inducing points for sparse Gaussian process classifier described below. The two obvious clusters appear due to single TCE and multiple detected TCEs systems, which is reflected in the maximum ephemeris correlation feature.

We would argue that Decision Tree ensembles over-fit the data, by pin pointing using random partitioning the feature space to classify labeled data points. For example, we can observe very high certainty for the predicted labels right at the intersection of the two classes, or in the middle of nowhere. This naturally leads to an argument, that machine learning builds a good space partitioning to classify the labeled points, but in our case might not necessarily generalize well, particularly when the clusters are so interconnected. For example, the brier score for Vespa derived probabilities is 0.11 on the labeled set, while both Random Forest achieve a score of 0.02, and logistic regression 0.06 - a minor sign of overconfidence. The reason for this argument is that, **vespa**, in comparison to machine learning, methods "knows" what it is looking at by having prior knowledge on the physical properties of the data, while machine learning manipulates the space of a labeled set.

We could use Random Forests to validate the hypothesis that it is in fact over fitting to outliers. Liu et al. (2008) proposed an outliers detection method based on random forests. For each data point on a tree we compute the path length P from the root node to its leaf node (class designation), i.e the number of the edges of the tree traversed. We also compute the number of the feature splits required to designate

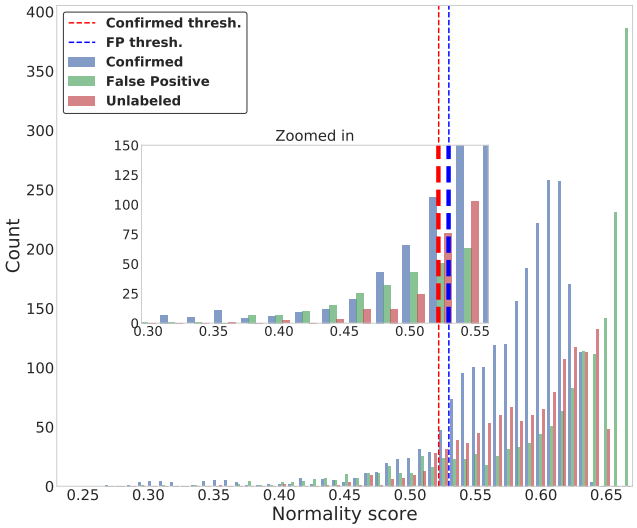

Figure 4. Normality scores histogram for FP, Confirmed instances in labeled data and for unlabeled data points. Scores were obtained using the average path length within each decision tree in random forest. Vertical lines correspond to selected cut-off value, below which data is considered to be an outlier. Zoomed in sub-figure is aligned appropriately with the threshold lines.

feature to a leaf node - depth D. Normality score N for the data instance i is defined as follows:

$$N_i = 2^{\frac{\mathbb{E}(D_i)}{\mathbb{E}(P_i)}} \tag{15}$$

We proceed by computing the scores separately for labeled data points in confirmed planets class, and FP, with histograms in Figure 4. We arbitrarily picked the threshold value, every point with normality below threshold was labeled as an outlier and removed followed by the models refit. Interestingly, if we compute the average absolute difference between **vespa** probabilities and random forest predicted probabilities, the average absolute difference for points found to be outliers was larger 0.27, and for the normal points 0.16. After removing the outliers, and running cross-validation once again on the labeled data, the recall score increased for all models, scoring as low as 0.95 for MLP and as high as 0.99 for random forest.

3.2 Probability comparison to Vespa

Given the findings above, we remove the outliers from the labeled dataset and perform cross-validation on a training set once again using every model, extracting the FP probabilities for each of the folds. For the unlabeled data FP probabilities we simply feed the data through the models trained on all of the labeled data. However, it is important to test if probabilities obtained correspond to actual model confidence. As per Niculescu-Mizil & Caruana (2005) maximum margin methods such as SVMs will push probabilities from 0 and 1, while classifiers with simplistic assumptions such as K-NN will push probabilities closer to 0 and 1, and MLP and decision tree ensembles would generally provide good probabilities. We can observe this in Table 2 that brier

Table 3. Probability comparison table with selected uncalibrated classifiers and Gaussian process classifier. Second and third column show average absolute difference for labeled, and unlabeled data. Remaining column stand for its skew for labeled, labeled FP and confirmed instances, and unlabeled data.

Classifier	Avg. Abs. Diff.		Skew of Abs. Diff.			
	Lab.	Unlab.	Lab.	FP	Conf.	Unlab.
Random Forest	0.14	0.26	2.08	1.15	3.66	1.01
Logistic Regression	0.15	0.31	2.02	1.15	3.33	0.73
LDA	0.15	0.31	1.76	1.32	2.73	0.75
Gaussian process	0.16	0.29	1.76	1.21	2.48	0.79

score for SVM models is much larger than for K-NN. We use isotonic regression Zadrozny & Elkan (2001) to investigate the reliability of the machine learning methods for probabilities prediction by once again cross-validating the algorithms over the labeled set, leaving one fold for isotonic regression calibration and one for prediction. Therefore, we have two probability results, one using cross-validation without calibration and one with isotonic calibration[5]. See Table 3 for the results of the selected uncalibrated classifiers.

We compare the results by comparing the mean absolute difference between the machine learning algorithm probabilities and **vespa** and its skew for labeled and unlabeled data. There is a slight improvement in mean absolute difference in probabilities for the SVM, K-NN, and boosting algorithms, and significant improvement in terms of the skew for unlabeled data after calibration, and as discussed in Niculescu-Mizil & Caruana (2005) does not affect significantly other models.

Random forest achieves the best alignment with **vespa** overall, as well as, on confirmed planets only. According to our findings, **vespa** and machine learning methods tend to agree more on confirmed planet probabilities in the labeled dataset, and tend to disagree more on the FP cases. In most cases where there is a significant disagreement between **vespa** and machine learning methods in labeled data on FP instances, machine learning models confidently predict a high probability for FP instances being FP, while **vespa** is quite uncertain in its predictions for some of these FP. This can be observed in Figure 7, where we plot Gaussian process probabilities against that of **vespa**.

Vespa or any physical modes are expected to produce better probabilities as the physical model *knows* what it is looking at. It might not be the case when the instance under consideration has light curve features that are not accounted for by the model. We find a number of examples where **vespa** does not provide probabilities which would align close to the actual disposition as discussed in the paragraph above. Investigation of types of these instances however is not possible as we do not have the labels for type of the false positive precisely as in McCauliff et al. (2014).

On the other hand, both calibrated and uncalibrated machine learning models are far less certain in their predictions for unlabeled dataset, see probability histogram in Figure 5. There are a number of possible explanations: either the FP scenarios considered in **vespa** are well represented in unlabeled dataset while under represented in the labeled dataset leading to machine learning models poor generalization. The latter part is particularly appealing given that in Figure 3 points in proximity to large FP cluster are particularly scattered, and could be suspected to appear on the boundary of the decision function learned by the classifiers.

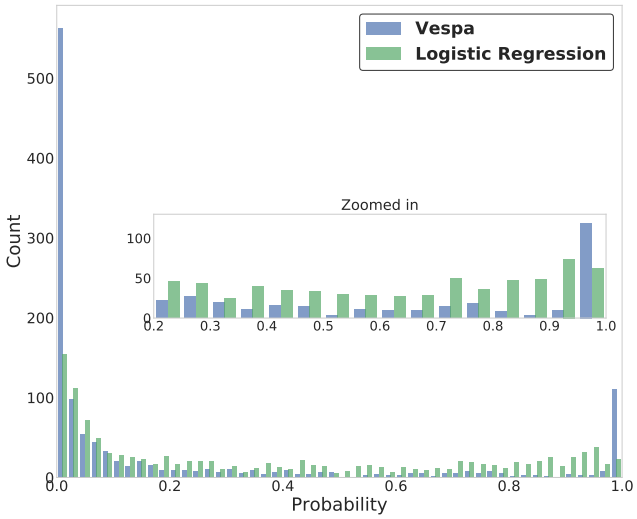

Figure 5. Histogram of VESPA and Logistic Regression derived probabilities for an instance to be FP in unlabeled dataset. Logistic Regression is significantly less confident in its predictions for unlabeled data compared to **vespa**.

Alternatively, we suspect that the structure of the unlabeled dataset may be different and therefore tested the normality of each point by using random forests as in the section above. First we trained the model on the labeled set and then estimated the normality for the unlabeled data, with the corresponding histogram in Figure 4. Given the normality histogram for the unlabeled dataset, there does not appear to be much novelty in the unlabeled dataset. That could be visually confirmed in Figure 3, as unlabeled points are distributed in proximity to the corresponding clusters, and the cluster means for labeled and unlabeled points are close when mapped using MDS.

3.3 Sparse Gaussian Process Classifier

As discussed in section 3.1 (and by Occam's razor) we would prefer simpler models over the more complicated, especially in the case like ours when the performance is only marginally better for more complicated models. Gaussian Processes (GP) allow us to enforce the simplicity of the decision boundary through specifying the covariance function, and employing fully Bayesian approach by specifying priors over the hyper parameters of the chosen covariance function. Using fully Bayesian approach would also provide as with the posterior distribution over the predicted probabilities, and ideally in the cases where **vespa** and GP predicted probabilities disagree the variance will be high, signaling the need for follow-up investigation of the TCE.

[5] See `https://github.com/jgamper/MScProject-Kepler-ML` for a full results table for calibrated and uncalibrated models.

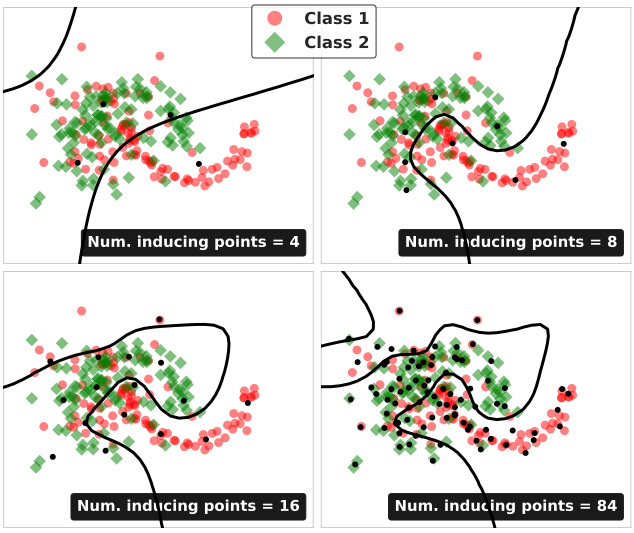

Figure 6. Sparse variational Gaussian process classification illustration of increasing complexity of the decision boundary on a simulated toy data, as the number of inducing points increases.

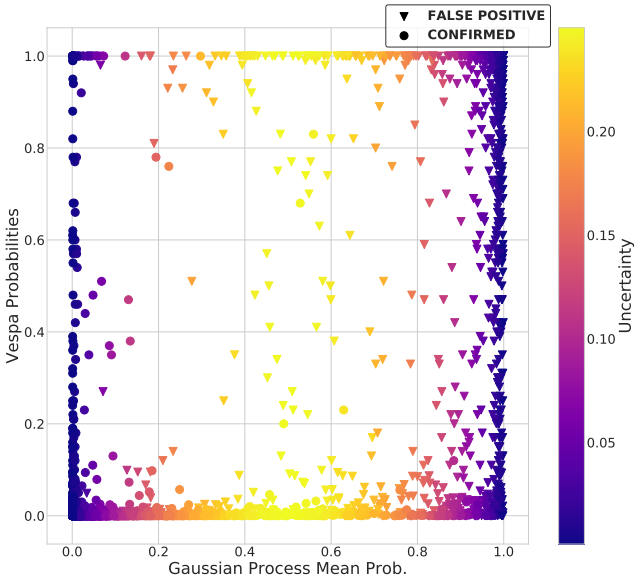

Figure 7. Vespa derived probabilities against Gaussian Processes derived probabilities colored by their variance respectively for the labeled dataset.

Advancements in stochastic variational optimization methods for sparse Gaussian processes, such as one developed in James et al. (2015), would easily scale to larger datasets of the future missions such as PLATO or NASA TESS, and is therefore tested in our work. In this GP model covariance is parameterized using M inducing, latent variables. The number of inducing points used further regularizes the GP, as illustrated in Figure 6. The advantage of parameterizing the sparse GP using M inducing points, where $M \ll N$ and N is the number of instances, is the reduction in algorithmic complexity from $O(N^3)$ down to $O(M^2N)$. Snelson & Ghahramani (2006) has shown that sparse GP parameterized with small number of inducing points, achieves performance comparable to full GP model. For an overview of spare GP models refer to Quiñonero Candela & Rasmussen (2005), and Titsias (2009) on optimisation of inducing points using variational inference. In our case we use 3 points obtained using K-Means algorithm at 10 different initializations (see Figure 3) for the GP to be most confident about the points appearing in high density space where a lot of training examples have been already obtained.

As in the methods above, we use cross-validation to derive probabilities and variance for labeled dataset, then train the GP model on the whole labeled dataset and extract probabilities for the unlabeled dataset. For the labeled dataset GP performs just as well as other models with AUC score of 0.98, and Brier score of 0.04. See Table 3 for absolute difference in probabilities with **vespa** and its skew - the results are similar to other machine learning models, with slightly higher disagreement between GP and **vespa** on confirmed planets. In Figure 7 we plot **vespa** and GP probabilities. Clearly a number of FP dispositions are missed by **vespa**. However, we must reiterate that Morton (2012) has used threshold of below 1% FP probability for a TCE to be considered a confirmed planet, therefore most of the points along the right side of the plot are still FP according to the Monte-Carlo based physical model, with 153 cases remaining misclassified.

If 1% probability threshold is adopted for machine learning models, 99% of FP cases are classified correctly in labeled dataset, compared to 92% by **vespa**. For confirmed planets machine learning models classify correctly between 74% and 92% correctly with 1% threshold. The Gaussian process classifier, manages to put a majority of misclassified planets into the area of higher uncertainty, and would give an indication of a need for follow-up investigation.

For the unlabeled dataset, GP achieved the same results as other machine learning methods, as in Figure 5 with far less certain predictions compared to **vespa**. Therefore, it would be particularly interesting to investigate the composition of FP instances in labeled dataset, and individual probabilities for each scenario within the **vespa** model for unlabeled data. If an imbalance is found between the compositions, under represented FP scenarios could be simulated as additional training data for machine learning models.

4 CONCLUSIONS

In this work, we have followed the procedures of McCauliff et al. (2014) to build a 9 dimensional representation of each TCE and tested a range of machine learning models in classifying and providing probabilities for confirmed and false positive dispositions of the planets. We found that there are outliers and point out that preference should be given to simpler models when all models perform reasonably well. By comparing the false positive probabilities of **vespa** and machine learning, we find that there is a possibility of under representation of certain false positives in the dispositioned data leading to significantly different probabilities on the unlabeled dataset. In future work, individual cases of false positives need to be investigated by looking into the probabilities of each FP scenario predicted by **vespa**, and additional training data simulated in case of imbalance. If adopt-

ing a 1% threshold (Morton 2012), most machine learning models and particularly Gaussian process capture 99% of FP, and up to 92% of confirmed planets. Those that are misclassified are assigned high probability variance, which aligns with the goal of the project to be able to confidently classify most of the planets and point out those that need further investigation. The flexibility of the Gaussian process model would allow us to further extend the power of the model once the class imbalance is investigated, by adopting specific kernel functions, and/or changing the number of inducing points. In particular, we can combine the powerful results of the random forest model and Gaussian process uncertainties by using a random forest kernel, as described in Davies & Ghahramani (2014).

Cross-validated results of the machine learning models show proof of concept that, in future missions such as PLATO and NASA's TESS, validation automation is feasible, leading to savings in science team efforts and more efficient allocation of follow-up observation resources based on probability uncertainties provided.

ACKNOWLEDGEMENTS

I would like to thank you D. J. Armstrong and T. Damoulas, who co-supervised my MSc project, extended their advice and support, and remained patient with me.

REFERENCES

Armstrong D. J., Pollacco D., Santerne A., 2016, Monthly Notices of the Royal Astronomical Society, 465, 2634
Batalha N. M., 2014, The Astrophysical Journal Supplement, 111
Bishop C. M., 2006, Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA
Borucki W. J., et al., 2003, Proceedings of the Conference on Towards Other Earthsand DARWIN/TPF and the Search for Extrasolar Terrestrial Planets, 22
Borucki W. J., et al., 2010, Science, 327
Borucki W. J., et al., 2011, The Astrophysical Journal, 736
Brown T. M., 2003, The Astrophysical Journal Supplement, 593
Bryson S. T., et al., 2013, Publications of the Astronomical Society of Pacific, 125
Burke C. J., et al., 2014, The Astrophysical Journal Supplement, 210
Davies A., Ghahramani Z., 2014, eprint arXiv:1402.4293
Diaz R. F., Almenara J. M., Santerne A., Moutou C., Lethuillier A., Deleuil M., 2014, Monthly Notices of the Royal Astronomical Society, 441
Fressin F., et al., 2011, Nature, 482
Friedman J. H., Tibshirani R., Hastie T., 2001, The elements of statistical learning
Gregory P. C., 2005, Bayesian Logical Data Analysis for the Physical Sciences
James H., Alexander G. d. G. M., Zoubin G., 2015, Proceedings of Machine Learning Research, 38
Jenkins J. M., Caldwell D. A., Borucki W. J., 2002, The Astrophysical Journal, 564
Jenkins J. M., et al., 2010, The Astrophysical Journal Letters, 713
Karttunen H., Kroger P., Oja H., Poutanen M., Donner K. J., 2007, Fundamental Astronomy
Koch D. G., et al., 2010, The Astrophysical Journal Letters, 713
Kohonen T., 1990, Proceedings of the IEEE, 78
Kyle C., Juan P., Gilles L., 2015, eprint arXiv:1506.02169
Liu F. T., Ting K. M., Zhou Z.-H., 2008, ICDM
Mandel K., Agol E., 2002, The Astrophysical Journal, 580
Mason J. W., 2008, Exoplanets Detection, Formation, Properties, Habitability. Springer-Praxis Books in Astronomy and Planetary Sciences
Matijevic G., Prsa A., Orosz J. A., Welsh W. F., Bloemen S., Barclay T., 2012, The Astronomical Journal, 143
Matthews Alexander G. d. G., van der Wilk M., Nickson T., Fujii K., Boukouvalas A., Leon-Villagra P., Ghahramani Z., Hensman J., 2016, eprint arXiv:1610.08733
McCauliff S. D., et al., 2014, The Astrophysical Journal, 806
Morton T. D., 2012, The Astrophysical Journal, 761
Morton T. D., Johnson J. A., 2011, The Astrophysical Journal, 738
Muirhead P. S., et al., 2012, The Astrophysical Journal, 747
Murray C. D., Correia A. C. M., 2010, Keplerian Orbits and Dynamics of Exoplanets
Neal R., 2008, n the proceedings of the PHYSTAT-LHC Workshop on Statistical Issues for LHC Physics
Niculescu-Mizil A., Caruana R., 2005, ICML '05 Proceedings of the 22nd international conference on Machine learning
Parviainen H., 2015, MNRAS, 450, 3233
Pedregosa F., et al., 2011, Journal of Machine Learning Research, 12, 2825
Quiñonero Candela J., Rasmussen C. E., 2005, J. Mach. Learn. Res., 6, 1939
Rogers S., Girolami M., 2011, A First Course in Machine Learning, 1st edn. Chapman & Hall/CRC
Sackett P. D., 1999, Planets Outside the Solar System: Theory and Observations
Schmidhuber J., 2015, Neural Networks, 61, 85
Snelson E., Ghahramani Z., 2006, in Advances In Neural Information Processing systems. MIT press, pp 1257–1264
Strobl C., Boulesteix A., Zeileis A., Hothorn T., 2015, Proceedings of the 18th International Con- ference on Artificial Intelligence and Statistics, 8
Struve O., 1952, The Observatory, 72
Thompson S. E., Mullally F., Coughlin J., Christiansen J. L., Henze C. E., Haas M. R., Burke C. J., 2015, The Astronomical Journal, 812
Titsias M. K., 2009, in In Artificial Intelligence and Statistics 12. pp 567–574
Torres G., et al., 2010, The Astrophysical Journal, Volume, 727
Winn J. N., 2014, Transits and Occultations. arXiv:1001.2010
Zadrozny B., Elkan C., 2001, ICML Proceedings of the Eighteenth International Conference on Machine Learning

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