Distinguishing Planetary Transits from Stellar Rotation Using a Random Forest Classifier

Rihaan Sodhani

9 October 2026

Abstract

Stellar rotational variability can mimic periodic transit signals and produce false-positive exoplanet detections when detrending fails to remove it completely. Such detections are grouped with other false positives, and existing classifiers often rely on large feature sets or deep neural networks that are difficult to interpret. This study investigated whether a random forest using a small set of interpretable features could distinguish planetary transits from stellar rotation. Simulated systems comprising rotation-only light curves and corresponding light curves with injected transits were cleaned, detrended with a Savitzky–Golay filter, and searched for periodic signals using the Box Least Squares algorithm. Ten features adapted from common diagnostics describing the depth, shape, noise, periodicity, and physical plausibility of the detected signal were extracted, yielding a balanced dataset of 1,030 light curves. A random forest was trained and evaluated using an 80:20 train-test split and cross-validation, achieving a moderate accuracy of 0.711±0.021, with a comparable precision, recall, specificity, and F1-score, and an area under the receiver operating characteristic curve of 0.790±0.022. This exceeded the threshold-based metrics, suggesting many misclassifications lie near the decision threshold, which could be adjusted based on classification priorities. Permutation importance showed that the number of transits and implied stellar density, both dependent on the detected period, were the most informative features, whereas shape-based features contributed little. These results support treating stellar rotation as a distinct classification problem. The classifier could serve as a pre-filter, reducing the number of rotation-induced detections requiring further vetting.

1. Introduction

In the decades since the first planets were found around other stars,[1,2] the number of confirmed exoplanets has exceeded 6,000.[3] Much of this growth comes from space-based photometric surveys, including Kepler,[4] K2,[5] and the Transiting Exoplanet Survey Satellite (TESS).[6] Since exoplanets are much fainter than their host stars, they are difficult to observe directly. Most detections use the effects that a planet produces on its host star. Detection methods include the transit photometry, radial velocity, direct imaging, astrometry, gravitational microlensing, and the pulsar timing method.[7] The majority of known exoplanets have been detected using the transit method.[3] A transit occurs when a planet passes in front of its host star along a telescope's line of sight and blocks a small fraction of its light. The resulting time series of stellar brightness, known as a light curve, contains information about the planetary system. Transits result in shallow, periodic dips in the light curve.

This method has certain limitations. Firstly, the chances of an appropriate orbital alignment are low. The method is also most sensitive to planets that produce relatively large photometric signals and transit frequently, favouring larger planets on short-period orbits.[7] Furthermore, a transit-like signal in a light curve is not sufficient to establish the presence of a planet. Periodic variations in stellar brightness can arise from astrophysical and instrumental phenomena, known as false positives if detected. The best-known astrophysical false positives are eclipsing binaries, a system of two stars orbiting a common centre of mass. A different source of variability arises from stellar rotation. As a star rotates, surface features such as starspots, which are cooler and emit less light than the surrounding photosphere, can periodically move into and out of view, producing a periodic modulation in the observed stellar brightness.[8] Though these variations must be removed from the light curve before searching for planets, the removal can be imperfect, potentially resulting in false positive detections.

The vetting of exoplanet candidates has traditionally relied on inspection. Checks include comparing the depths of odd and even transits, searching for secondary eclipses, and examining transit shape. These approaches are slow and require experts, resulting in bottlenecks as surveys like TESS produce thousands of candidates.[9] Machine learning (ML) techniques have recently been explored as an alternative approach to identify false positive exoplanet candidates. It can be formulated as a supervised binary classification task, in which the input is derived from the light curve and the output is whether the curve corresponds to an exoplanet or not.

2. Background

A seminal work introduced Box Least Squares (BLS) periodograms as an algorithm to search for periodic transits in stellar photometric time series, modelling transits as box-like dips. By fitting this box-shaped model over a range of trial periods and transit durations, the algorithm identifies the period that provides the strongest fit to a transit-like signal.[10]

Recent research has increasingly explored machine-learning techniques for exoplanet detection. One study removed low-frequency stellar variability such as starspot rotation from TESS data using a Savitzky–Golay filter. The detrended light curves were then classified using a 1D convolutional neural network trained to distinguish planetary candidates from false positives, including variable stars and eclipsing binaries.[11] Another paper used random forests to automate the classification of transit-like signals in Kepler data using 77 features derived from the traditional pipeline. Based on the principle of parsimony, the authors argued that classifiers using fewer features are preferable to complex models.[9] A recent study developed a Vision Transformer to detect exoplanets and estimate their orbital parameters from radial velocity (RV) time-series spectra. The scarcity of confirmed exoplanets limited the availability of labelled ground-truth data. The authors addressed this limitation by injecting synthetic Keplerian planetary signals into solar data from the NEID spectrograph.[12]

Existing approaches have largely treated stellar rotation as a preprocessing problem rather than a classification problem. It is typically suppressed by filtering the light curve before the transit search. In practice, rotational variability can survive detrending and be detected as a periodic signal. Such detections are grouped with eclipsing binaries and other sources into a single false-positive class. Moreover, existing classifiers often rely on large feature sets or deep neural networks, which are difficult to interpret. To address these gaps, this study aims to use random forests with a small, interpretable feature set to distinguish transits from stellar rotation after detrending and BLS analysis.

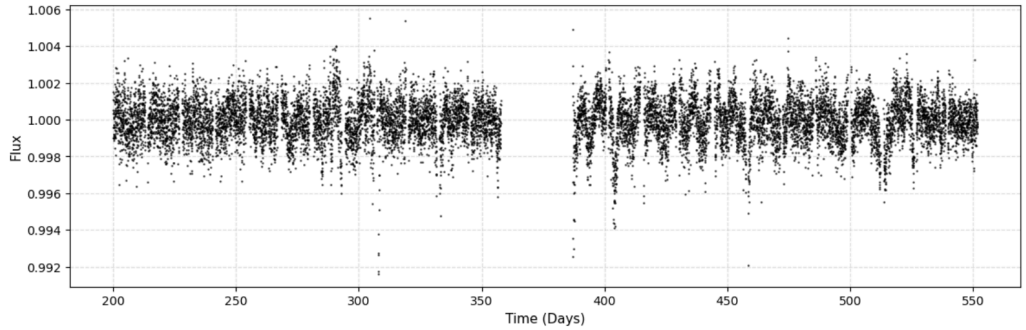
3. Methodology

3.1 *Light curve data*

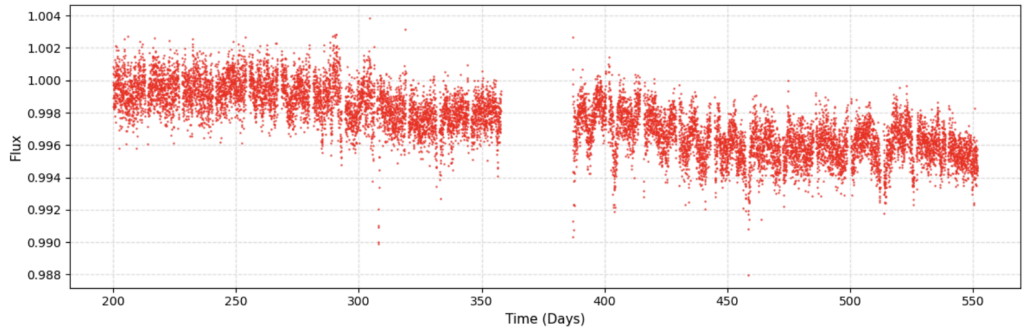
The light curves used in this study were simulated and provided by the lightcurve.ai team at TIFR as NumPy compressed archives (.npz).[13] The dataset consists of simulated systems. Each system contains a time array and two normalised flux arrays: a light curve representing stellar rotation and a corresponding light curve containing injected planetary transit signals. The simulations include noise and data gaps to resemble real observations. All light curves share the same time sampling, spanning approximately 352 days at a cadence of approximately 30 minutes.

An example system is shown in Figure 1. Subtracting the rotation flux from the transit flux cancels the shared noise and variability. This contains shallow transits that are distinguishable by eye. However, the classifier must receive both original noisy light curves, preserving the observational noise and stellar variability, thus providing two labelled classes with known ground truth.

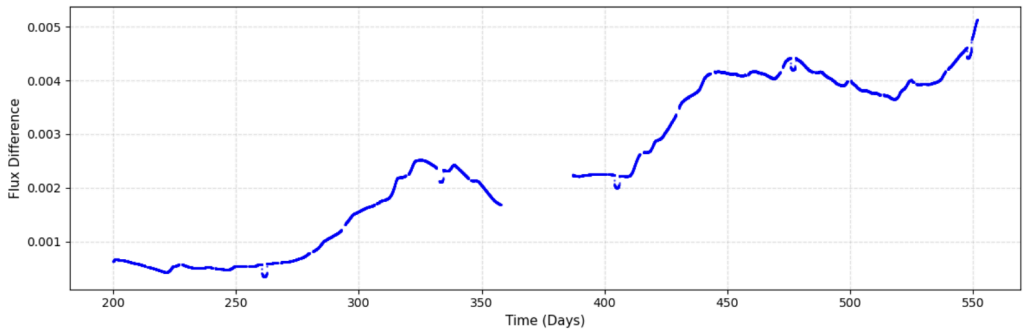
The use of simulated data provides several advantages. Firstly, because each light curve is generated with a known signal, its class label is exact, whereas labels for real survey data can be uncertain.[11] Secondly, simulation makes it possible to produce a large labelled dataset, which would otherwise be limited by the small number of confirmed planets. Furthermore, the two classes are balanced by construction, minimising potential model bias caused by rare detections. The main trade-off is that simulated light curves may not reproduce all the instrumental and astrophysical effects present in real data.[12]



(a) Transit flux data



(b) Stellar-Rotation flux data



(c) Difference between transit and stellar-rotation flux

Figure 1. Simulated light curves containing transit and stellar-rotation flux data. (a) Normalised flux measurements containing injected planetary transit signals. (b) Flux measurements representing stellar rotational variability. (c) Difference between the transit and stellar-rotation flux measurements.

3.2 *Light curve preprocessing*

Each light curve was preprocessed using the same sequence of steps. Firstly, the flux array was converted to a LightCurve object using the LightKurve library.[14] Secondly, cadences with missing (NaN) flux values were removed. Outliers were then removed by sigma clipping, with points more than 4σ above the median flux and more than 20σ below it being discarded. The asymmetric thresholds remove only extreme negative values, so that shallow downward dips, including transits, are preserved.[15]

Finally, the light curves were detrended to suppress variability on timescales longer than a transit. A Savitzky–Golay filter was used to fit a low-order polynomial to the flux within a window moving along the light curve,[16] estimating a smooth trend. The flux was divided by

this trend, flattening the data. Hence, no separate normalisation step was required. No transit mask was applied as the transit times are not known in advance. A window of 401 cadences was used, corresponding to approximately 8.4 days at the 30-minute cadence. An odd number was chosen so that the window has a well-defined centre. Since transits last a few hours, detrending leaves the transit intact. Ideally, slow stellar variability is removed, but the detrending is imperfect. For example, stellar-rotational modulation with shorter periods or sharp features is only partly captured by the trend and survives into the flattened light curve.[15]

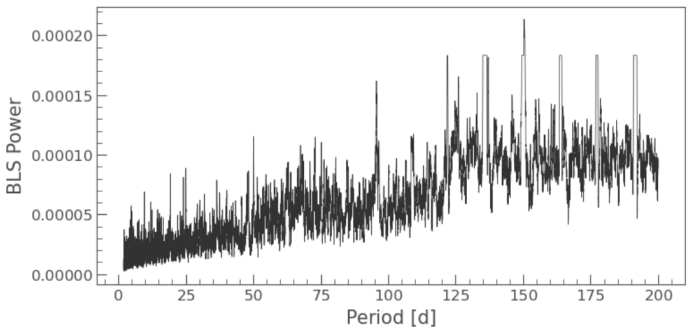
3.3 *Periodic Signal Search*

Each flattened light curve was searched for periodic transit-like signals using the Box Least Squares (BLS) algorithm,[10] as implemented in Lightkurve via Astropy.[17] For each trial period P and trial duration D, it found the mid-transit time t_0 of the box that best fit the data, assigning that combination a power measuring how much the box model improves the fit relative to a flat line. Repeating this over a grid of trial values gives a periodogram, a curve of BLS power against period, whose highest peak marks the most transit-like periodic signal in the data.[10]

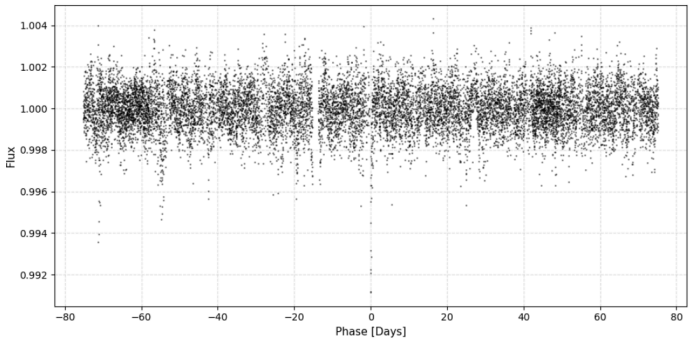
Trial periods were taken from a grid of 100,000 values spaced uniformly between 2 and 200 days. Using the recovered P and t_0, each light curve was phase folded. Each time t was converted to a phase ϕ measured in days from mid-transit, using the equation

$$\phi = \left[\left(t - t_0 + \frac{P}{2}\right) mod\ P\right] - \frac{P}{2},$$

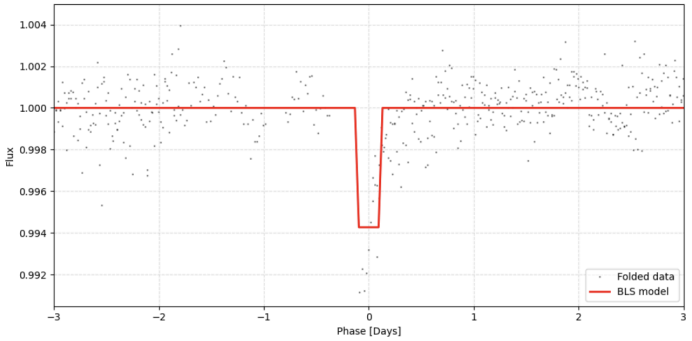
so that all transits overlap near $\phi = 0$ and phase ranges from $-\frac{P}{2}$ to $+\frac{P}{2}$. An example for a light curve containing a transit is shown in Figure 2.



(a) Transit flux data



(b) Phase-folded light curve



(c) BLS model

Figure 2. BLS periodogram and phase-folded light curve containing a transit. (a) BLS power as a function of trial orbital period. (b) Light curve phase-folded at the BLS-derived period. (c) Zoomed-in view of the phase-folded light curve with BLS model overlaid.

3.4 *Feature extraction*

For each light curve, ten features were computed from the flattened, phase-folded light curve using the BLS period P, mid-transit time t_0 and duration D. The features, described in Table 1, are adapted from diagnostics used in transit candidate vetting.[18,19] The features are based on three windows defined using phase ϕ relative to the detected duration D: the core window covering the central part of the transit, defined as $|\phi| < 0.3D$; the full window covering the entire transit, defined as $|\phi| < 0.5D$; and the out-of-transit window, defined as $D < |\phi| < 4D$. The gap between 0.5D and D excludes the transit edges from the baseline.

Table 1. Features extracted from each light curve.

No.	Feature	Definition	Explanation
1	Depth (ppm)	median(F_out) − median(F_core)	A measure of how much light is blocked.
2	Duration (h)	D from BLS	A measure of how long the transit dip lasts.
3	Number of transits	Number of distinct epochs with points in the full window	How many times the dip was observed.
4	Out-of-transit scatter (ppm)	std(F_out)	A measure of how noisy the light curve is around the transit
5	Signal-to-noise ratio	(δ_full / σ_out) × √N_full	A measure of How strongly the dip stands out above the noise
6	V-shape ratio	δ_core / δ_full	A measure of whether the bottom dip is flat or pointed.
7	Odd-even difference (σ)	|δ_odd − δ_even| / (σ_out / √(N_core/2))	A measure of whether alternate dips have the same depth
8	Secondary depth (ppm)	Depth within 0.5D of $\|\phi\|$ = P/2	A measure of whether the star also dims halfway between transits.
9	Implied stellar density (g cm^{-3})	$3\pi(a/R^*)^3 / (GP^2)$, with $a/R^* = P/(\pi D)$	The implied density of the host star, calculated using Kepler's third law, assuming a central transit and a circular orbit.[20]
10	Depth scatter (ppm)	std(δ_n) over individual epochs n	A measure of whether the dip has the same depth at every transit.

3.5 *Dataset*

The extracted features were stored as a table in a CSV file, with one row per light curve. Each row contained 17 columns, including the 10 extracted features, forming the input to the classifiers, class labels, and other metadata such as the source file name. A class label of 1 denotes a light curve containing injected transits, whereas 0 denotes a rotation-only light curve.

Of the 1,498 light curves processed (749 systems), 468 rows (31.2%) contained at least one undefined (NaN) feature, arising where a feature cannot be computed. NaN values occurred in four of the ten features: odd-even difference, depth scatter, transit depth, and V_shape. The most frequent was the odd-even difference, which was undefined in 466 rows. This occurs

when either the odd- or the even-numbered transits contain fewer than three points within the core window, so that one of the two depths cannot be measured. The depth scatter was undefined in 421 rows, which occurs when fewer than two individual transits contain at least three core-window points. The transit depth and the V-shape ratio were each undefined in two rotation-only rows. All rows containing at least one NaN feature were excluded. The excluded fraction was similar for the two classes, at 33.6% of rotation-only light curves (252) and 28.8% of transit light curves (216), so exclusion did not introduce a substantial class imbalance. The final dataset contained 1,030 light curves, of which 533 (51.7%) contained injected transits (label 1) and 497 (48.3%) were rotation-only (label 0). Thus, the classes were balanced.

The dataset was split into training (80%) and test (20%) sets by system, so that the transit and rotation-only light curves of a given system were always assigned to the same set. Since the two versions share the same noise and stellar variability, this prevents the model from recognising the shared signal instead of learning to distinguish transits from rotation.

3.6 *Classification Model*

Light curves were classified using a random forest implemented in scikit-learn.[21] A random forest is an ensemble learning method that combines the predictions of multiple decision trees, each trained on a randomly sampled subset of the training data and considering a random subset of features at each split. Each tree classifies a light curve through a sequence of binary splits, at each of which a single feature is compared with a threshold, until a terminal node, or leaf, assigns a class. Finally, the individual trees vote on the predicted class, with the majority vote determining the final prediction.[22]

Several properties make a random forest well suited to this problem. Firstly, it is interpretable. Each feature is a physically meaningful property of the detected signal, and the contribution of each feature to the classification can be measured directly, so the model shows which physical properties distinguish transits from stellar rotation, unlike opaque classifiers such as neural networks. Secondly, it is resistant to overfitting, which matters for a modest, noisy dataset of around a thousand light curves. Furthermore, they can handle features combining ratios, physical quantities, and statistical measures. Since each split compares a single feature with a threshold, these require no rescaling. Finally, it captures non-linear relationships, suiting several features that indicate a transit only within a range of values.

The forest consisted of 500 trees, each considering 3 randomly selected features at every split. A minimum of two samples was required in each leaf, which prevents trees from using individual light curves and limits overfitting. Class weights inversely proportional to class frequency were applied, although the balanced dataset means their effect is small. A light curve was classified as containing a transit when this probability exceeded 0.5. All other hyperparameters were left at their scikit-learn defaults, and a fixed random seed (random state 42) was used to make the results reproducible.

3.7 *Classifier Performance*

The accuracy of a classifier is calculated as the fraction of all light curves classified correctly. However, it does not distinguish between the different types of classification errors. For the light curves in a test set, a true positive (TP) is a transit light curve correctly classified as a transit, and a true negative (TN) is a rotation-only light curve correctly classified as rotation. A false positive (FP) is a rotation-only light curve classified as a transit, and a false negative (FN) is a transit light curve classified as rotation. These were represented using a confusion matrix. Precision, recall, specificity, and the F1-score were calculated using the matrix, using cross validation across different train-test splits to obtain uncertainty.

Precision is the fraction of light curves classified as transits that truly contain one, measuring how often rotation is mistaken for a transit. Recall measures the proportion of genuine transit light curves that are successfully identified, making it a measure of the model's ability to detect planetary signals. Specificity measures the proportion of rotation-only light curves correctly identified, quantifying the model's ability to avoid false detections caused by stellar variability. The F1-score, defined as the harmonic mean of precision and recall, was also calculated to provide a single measure of the model's ability to identify transit signals while limiting false detections.

3.8 *Receiver Operating Characteristics Analysis*

The performance of the random forest may be influenced by a poorly calibrated classification threshold. To assess classifier performance independently of the selected classification threshold, receiver operating characteristic (ROC) curve was created, plotting the true positive rate (recall) against the false positive rate (1–specificity) as the threshold is varied from 0 to 1.[23] The area under the curve (AUC) was subsequently computed.

3.9 *Feature Importance*

Permutation importance was used to assess the contribution of each feature to the classifier's predictions. The values of one feature were randomly shuffled, breaking its relationship with the class label, while all other features and the trained model were left unchanged. The mean resulting decrease in ROC-AUC was taken as that feature's importance. The importance of each of the ten features was plotted.

4. Results and Discussion

4.1 *Classifier Performance*

Taking exoplanet as the positive class (1) and stellar rotation as the negative class (0), the confusion matrix for the random forest classifier is shown in Figure 3. The classification metrics are presented in Table 2. The classifier achieved a mean accuracy of 0.711±0.021, well above the value of 0.5 expected from random guessing for two balanced classes. The metrics lie in a narrow range, from a recall of 0.699±0.038 to a precision of 0.730±0.017, so

the classifier performs similarly regardless of which aspect of performance is measured. The standard deviations are small (0.017–0.038), indicating that performance is stable across different divisions of the data and does not depend on a particular choice of training and test systems.

The confusion matrix shows that both types of error, false positives and false negatives, occur at comparable rates. Of the light curves shown, 24 rotation-only light curves were misclassified as transits and 31 transit light curves were misclassified as rotation. The same pattern appears in the cross-validated metrics, where specificity (0.723±0.028) slightly exceeds recall (0.699±0.038). As the difference is within one standard deviation, the classifier shows no strong bias towards either class.

Overall, the classifier separates the two classes substantially better than chance, but not reliably enough to be used alone. In a vetting application, it could reduce the number of rotation-induced detections passed on for inspection, but candidates it accepts would still require further checks.

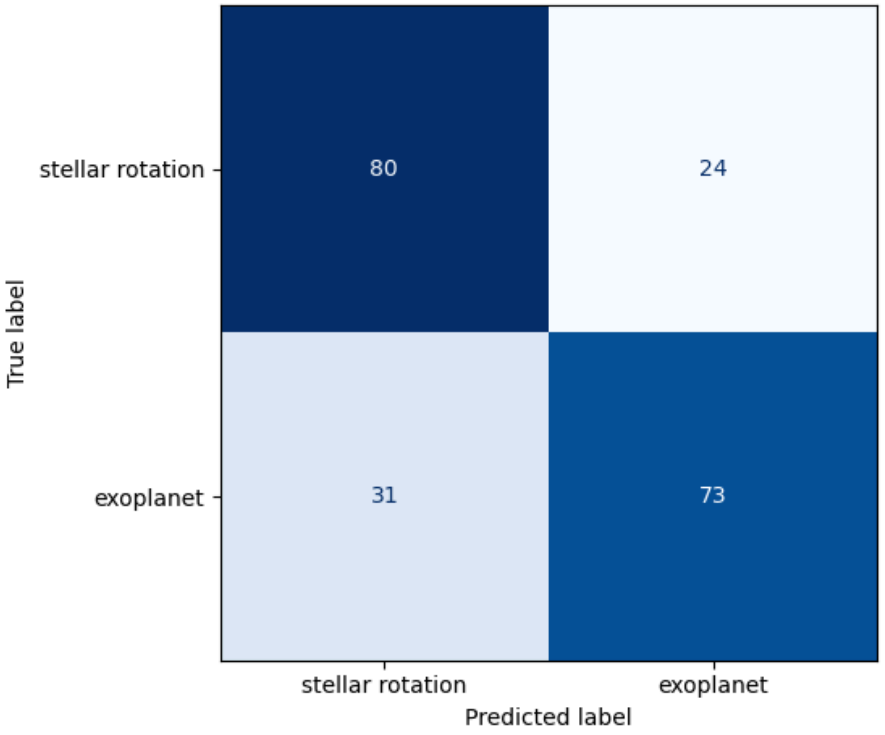


Figure 3. Confusion matrix for the random forest classifier. Each matrix shows the number of true positives (bottom right), true negatives (top left), false positives (top right), and false negatives (bottom left).

Table 2. Classification performance metrics for the random forest classifier.

No.	Metric	Value
1	Accuracy	0.711±0.021
2	Precision	0.730±0.017
3	Recall	0.699±0.038
4	Specificity	0.723±0.028
5	F1-score	0.714±0.022

4.2 *Receiver Operating Characteristics Analysis*

The ROC curve of the random forest classifier is shown in Figure 4. The AUC was found to be 0.790±0.022. This means that a randomly chosen transit light curve is assigned a higher transit probability than a randomly chosen rotation-only light curve about 79% of the time, compared with 50% for a classifier that guesses. The small standard deviation indicates that this performance is stable across different divisions of the data.

The AUC exceeds every metric based on the threshold of 0.5 in Table 2. This suggests that many classification errors are near misses, with probabilities close to the threshold, rather than confident misclassifications. Consequently, even a small change in the threshold would reclassify a noticeable number of light curves. Lowering the threshold would convert some missed transits into detections, at the cost of potentially misclassifying rotation-only light curves, whereas raising it would have the opposite effect. Hence, the appropriate threshold therefore depends on the purpose for which the classifier is used. A lower threshold is appropriate when missing an exoplanet is more costly than examining a false candidate. This applies when the classifier serves as an early pre-filter ahead of more rigorous vetting, so that rotation-induced candidates it accepts can still be rejected later. A higher threshold is appropriate when each accepted candidate carries a significant cost. For example, selecting targets for detailed study, such as atmospheric characterisation, demands a sample with few false positives. Given the moderate accuracy of this random forest classifier, a lower threshold is likely the preferable option.

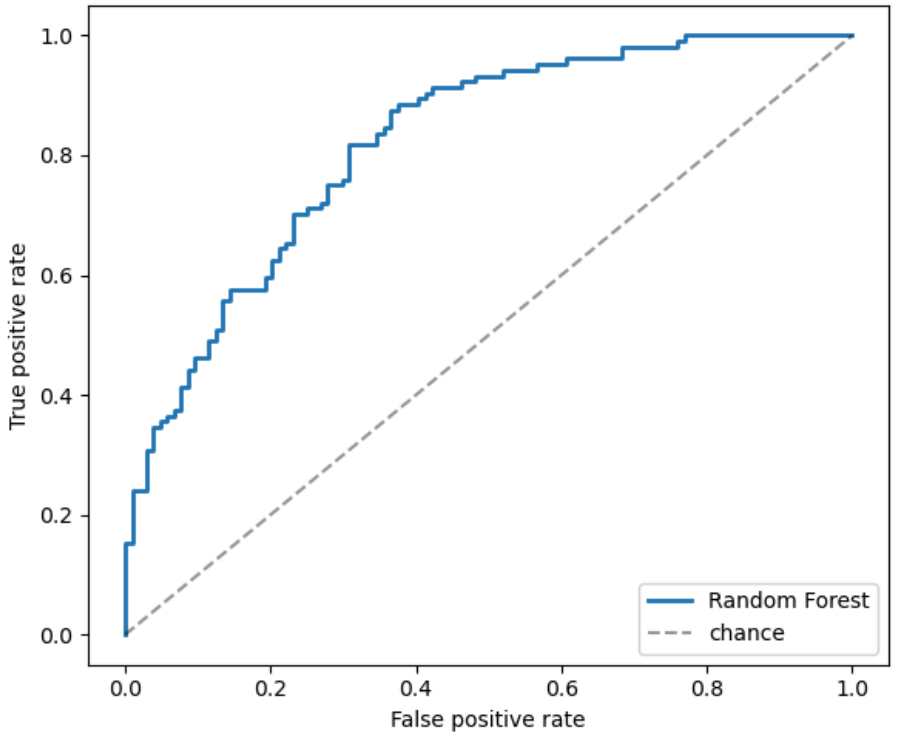


Figure 4. Receiver operating characteristic (ROC) curve for the random forest classifier. The line y=x, representing a random classifier, is shown as a grey dashed line.

4.3 *Feature Importance*

The permutation importance of each of the ten features is shown in Figure 5. The number of transits was the most important feature: shuffling it reduced the AUC by 0.05844. The duration was the least important by a large margin, with almost no effect (0.00022). However, its negligible importance does not mean it is uninformative. Since duration is used to calculate stellar density, when shuffled, much of its information remains available through this feature. Following the principle of parsimony, this feature could be excluded to simplify the model. Even the most important feature reduced the AUC by less than 0.06 from 0.790±0.022, so no single feature is indispensable, and the classifier draws on information spread across several features. This indicates a robust classifier that can distinguish planetary transits from stellar variability more effectively than a single diagnostic.

The two most important features, the number of transits and the implied stellar density (0.04216), both describe the period of the detected signal. A planetary transit is strictly periodic, and its period and duration are related through Kepler's third law. On the other hand, rotational modulation may be only quasi-periodic as starspots evolve over time, so brightness variations may not repeat exactly. Therefore, features that test periodicity likely address a genuine distinction between the classes. By contrast, the shape-based features, the V-shape ratio and the odd-even difference, contributed little. This may reflect the nature of the classification task as well. The V-shape ratio and odd-even difference can help identify

eclipsing binaries, whose eclipse profiles or alternating eclipse depths may differ from those of planetary transits. However, this study distinguishes planetary transits from pure stellar rotational variability, which may not exhibit these characteristic features.

The statistical features, namely the out-of-transit scatter (0.01277), the signal-to-noise ratio (0.01288), and the depth scatter (0.01129), were of moderate importance. The contribution of the out-of-transit scatter was not anticipated, since the two simulated light curves of each system share the same noise patterns. However, the intermediate importance of depth scatter and signal-to-noise ratio is consistent with the expectation that rotation-induced dips vary between epochs as starspots evolve.

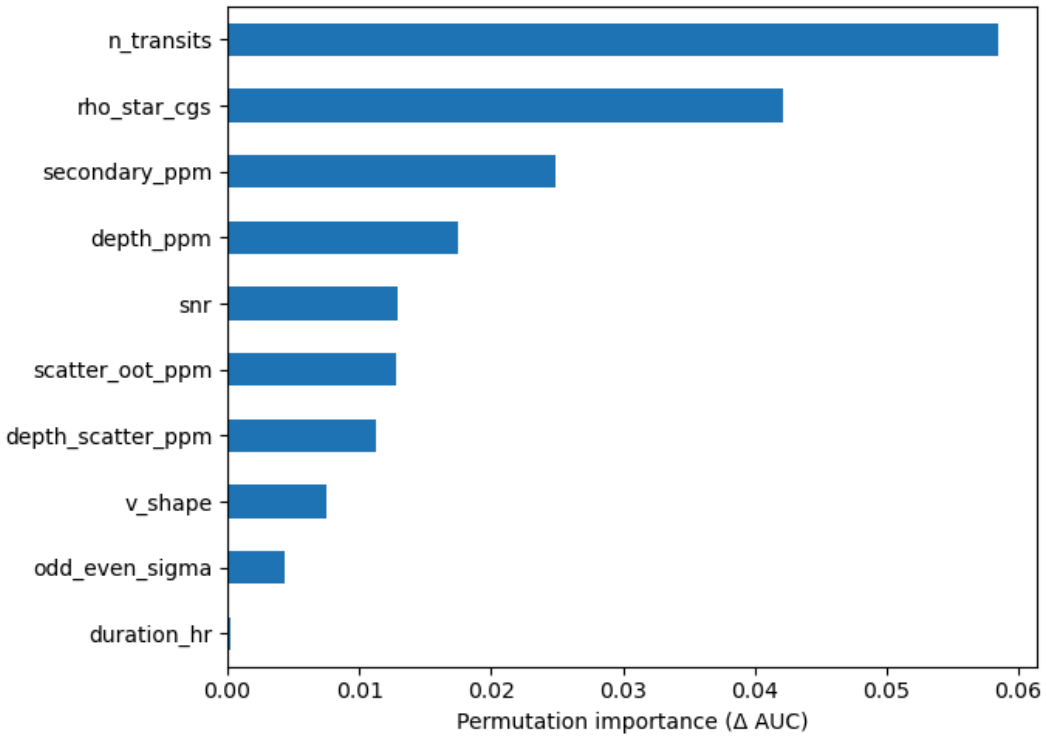


Figure 5. Permutation importance of features in the Random Forest classifier, measured by the decrease in ROC-AUC.

5. Conclusion

This study aimed to determine whether a random forest classifier using a small set of physically interpretable features could distinguish planetary transits from stellar rotational variability that survives detrending. Simulated light curves were used, each system comprising a rotation-only light curve and a corresponding light curve containing injected transits. Each light curve was cleaned, detrended using a Savitzky–Golay filter, and searched for periodic signals using the BLS algorithm. Ten features describing the depth, shape, periodicity and physical consistency of the detected signal were then extracted. After excluding light curves with undefined features, a random forest was trained and evaluated on

1,030 light curves. The classifier achieved an accuracy of 0.711 ± 0.021 and an ROC-AUC of 0.790±0.022, with a comparable precision, recall, specificity, and F1-score. The most important features were the number of transits and implied stellar density, which depend on the period of the detected signal, whereas the shape-based features contributed little.

These results show that stellar rotation can be addressed as a distinct classification problem, not solely as a preprocessing step, and that a compact set of physically motivated features carries substantial information for separating the two classes. Because the features correspond to measurable physical properties associated with common diagnostic tests for candidate vetting, the classifier's decisions are interpretable. Although its performance is not sufficient for the classifier to be used alone, its output allows the threshold to be adjusted to the application. With a lower threshold, it could serve as a pre-filter that reduces the number of rotation-induced detections requiring further vetting.

Certain limitations remain. Firstly, the light curves were simulated and may not reproduce all the instrumental and astrophysical effects present in real data. The transit and rotation-only light curves of each system shared identical noise and variability, which does not reflect observed survey data. Furthermore, genuine transits are rare, introducing class imbalance in real data. Thus, future work applying the methodology investigated in this study to observed light curves from missions such as Kepler and TESS could test whether such a classifier generalises beyond the simulated dataset. Secondly, the classifier's accuracy of 0.711 is insufficient for it to be used alone. Additional features could improve performance, and comparison with alternative machine-learning models, such as gradient-boosted trees or convolutional neural networks could identify whether ML-based approaches can distinguish planetary transits from stellar rotational variability without the need for further analysis. Finally, this study addressed only stellar rotation. Other signals that are conventionally suppressed during preprocessing, such as instrumental systematics, can also survive detrending and produce spurious detections.[19] Future studies could apply a similar classification approach to these signals.

Acknowledgements

I thank the Lightcurve.ai team at the Tata Institute of Fundamental Research (TIFR) for providing the simulated light curve data used in this study. I am grateful to Professor Shravan Hanasoge and Avik Dasgupta for their guidance and support throughout the project. This work received no external funding.

I acknowledge the use of Claude (Anthropic) during the preparation and revision of this research report. In the final editing stage, it was used regularly to review grammar, phrasing, sentence structure, and clarity. Following the use of any AI tool, I reviewed, edited, and confirmed the scientific accuracy of all the generated text. All experimental design, data analysis, interpretations, and conclusions are my own.

References

1. Wolszczan, A. & Frail, D. A planetary system around the millisecond pulsar PSR1257 + 12. Nature 355, 145–147 (1992).

2. Mayor, M. & Queloz, D. A Jupiter-mass companion to a solar-type star. Nature 378, 355–359 (1995).

3. Christiansen, J. L. et al. The NASA Exoplanet Archive and Exoplanet Follow-up Observing Program: Data, Tools, and Usage. Planet. Sci. J. 6, 186 (2025).

4. Borucki, W. J. et al. Kepler planet-detection mission: Introduction and first results. Science 327, 977–980 (2010).

5. Howell, S. B. et al. The K2 Mission: Characterization and Early Results. Publ. Astron. Soc. Pac. 126, 398–408 (2014).

6. Ricker, G. R. et al. Transiting Exoplanet Survey Satellite. J. Astron. Telesc. Instrum. Syst. 1, 014003 (2015).

7. Wright, J. T. & Gaudi, B. S. Exoplanet detection methods. In Planets, Stars and Stellar Systems 3, 489–540 (Springer, 2013).

8. Aigrain, S., Pont, F. & Zucker, S. A simple method to estimate radial velocity variations due to stellar activity using photometry. Mon. Not. R. Astron. Soc. 419, 3147–3158 (2012).

9. McCauliff, S. D. et al. Automatic classification of Kepler planetary transit candidates. Astrophys. J. 806, 6 (2015).

10. Kovács, G., Zucker, S. & Mazeh, T. A box-fitting algorithm in the search for periodic transits. Astron. Astrophys. 391, 369–377 (2002).

11. Rao, S. et al. Nigraha: machine-learning-based pipeline to identify and evaluate planet candidates from TESS. Mon. Not. R. Astron. Soc. 502, 2845–2858 (2021).

12. Gavankar, A., Mittal, T., Ninan, J. P. & Hanasoge, S. Machine learning for radial velocity analysis. I. Vision transformers as a robust alternative for detecting planetary candidates. Astron. J. (2026).

13. Harris, C. R. et al. Array programming with NumPy. Nature 585, 357–362 (2020).

14. Lightkurve Collaboration et al. Lightkurve: Kepler and TESS time series analysis in Python. Astrophysics Source Code Library, ascl:1812.013 (2018).

15. Hippke, M., David, T. J., Mulders, G. D. & Heller, R. Wōtan: Comprehensive time-series detrending in Python. Astron. J. 158, 143 (2019).

16. Savitzky, A. & Golay, M. J. E. Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. 36, 1627–1639 (1964).

17. Astropy Collaboration et al. Astropy: A community Python package for astronomy. Astron. Astrophys. 558, A33 (2013).

18. Coughlin, J. L. et al. Planetary candidates observed by Kepler. VII. The first fully uniform catalog based on the entire 48-month data set (Q1–Q17 DR24). Astrophys. J. Suppl. Ser. 224, 12 (2016).

19. Thompson, S. E. et al. Planetary candidates observed by Kepler. VIII. A fully automated catalog with measured completeness and reliability based on Data Release 25. Astrophys. J. Suppl. Ser. 235, 38 (2018).

20. Seager, S. & Mallén-Ornelas, G. On the unique solution of planet and star parameters from an extrasolar planet transit light curve. Astrophys. J. 585, 1038–1055 (2003).

21. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011).

22. Breiman, L. Random forests. Mach. Learn. 45, 5–32 (2001).

23. Fawcett, T. An introduction to ROC analysis. Pattern Recognit. Lett. 27, 861–874 (2006).