

Comparative Investigation of Machine Learning Model Performances for Exoplanet Transit Detection Using Kepler Data

Syed Aarsal Abbas, Andrew James Wildbridge
arsalabbassyedd@gmail.com, awildrid@purdue.edu

ABSTRACT

Mankind has been fascinated with discovering new places, and exoplanet detection is no outlier. Exoplanets are planets that are located outside our solar system. The massive volume of data on these exoplanets, gathered by the National Aeronautics and Space Administration (NASA) missions like Kepler and Transiting Exoplanet Survey Satellite (TESS) on exoplanets, calls for automated classification. This investigation compares how each machine learning model performs for exoplanet detection using data gathered from the Kepler Space Telescope, focusing on three models: logistic regression, random forest, and shallow neural network. The performance metrics in this study are accuracy, precision, recall and F_1 score.

The training, validation, and testing of all three models was performed on the same Kepler dataset. The models received four statistical features as input, derived from the star's light curve (a graph of luminous flux against time): mean flux, standard deviation of the flux, maximum flux and minimum flux. The performance of each model was assessed using evaluation metrics of accuracy, precision, recall and F_1 score derived from confusion matrices generated in Python.

The data on which the models were trained were derived from transit photometry, where periodic dips in the brightness of a star indicate an orbit. These dips can be seen in light curves.

This investigation demonstrates that machine learning can be effectively applied to exoplanet detection. This investigation also highlights the importance of model selection, feature representation and evaluation metrics in classification tasks.

Keywords: Exoplanet, Machine Learning, Astrophysics, Computer Science, Python, Logistic Regression, Random Forest, Light Curve, Neural Network, Transit Detection.

INTRODUCTION

The Universe has countless galaxies, each having billions of stars, many of which have their own exoplanets. In our search for the possibility of life beyond our solar system, scientists have discovered more than 6,000 planets [1]. These discoveries, largely aided by technological advancements in computational techniques [2], confirm that there are abundant planetary systems exhibiting planets with a wide range of sizes, compositions, and orbital characteristics across the galaxy [3].

The first exoplanet orbiting a main sequence star was 51 Pegasi b, discovered in 1995 by Mayor and Queloz. [4]. This discovery provided the first direct evidence that there exist other planetary systems. . The search for exoplanets rapidly became an active domain for scientific research, and one equally quickly supported by funding authorities [5].

Exoplanet detection is of high interest to the broad scientific community for several reasons. It enables us to learn and analyze how planets form by studying their composition and orbital dynamics, and it can teach us about how interplanetary conditions could differ. For example, 55 Cancri e is assumed to contain a large mass of diamond due to its high temperatures and composition [6]. Furthermore, exoplanet detection contributes to the search for worlds potentially habitable by mankind [7].

Traditionally, methods used to identify exoplanets involve a significant amount of human labor to determine whether an exoplanet is real or a false positive, which is a result that suggests the presence of an exoplanet, but upon further investigation there is none. However, this is very time-intensive and is prone to frequent human error due to the massive amounts of data acquired from missions and the numerous planets each mission discovers [8].

The Kepler Mission, a National Aeronautics and Space Administration (NASA) Discovery mission launched on March 6, 2009, was the first space mission dedicated to the search for Earth-sized and smaller planets in the habitable zone of other stars in our neighborhood of the galaxy [9]. The Transiting Exoplanet Survey Satellite (TESS) was launched in 2018 to search for planets transiting bright and nearby stars [10]. When a planet passes in front of its parent star, as seen from our solar system, it blocks a small fraction of the light from that star; this is known as a transit [11]. Both missions have transformed exoplanetary science by providing large-scale and high-quality data that are suitable for computational analysis.

The transit method has become the most successful detection method, surpassing the combined detection counts of all other methods [12]. This method involves measuring the brightness (flux) of a star over a long period of time. If an exoplanet is orbiting the star, eventually the planet will come between the space-based observatory and the star. This will be observed as a very tiny drop in the brightness of the star

as the exoplanet blocks some of the star's light. In a light curve, a graph of the star's luminous flux over time, there will be a periodic dip in the brightness due to the planet. A visual example of this is shown in Fig. 1.

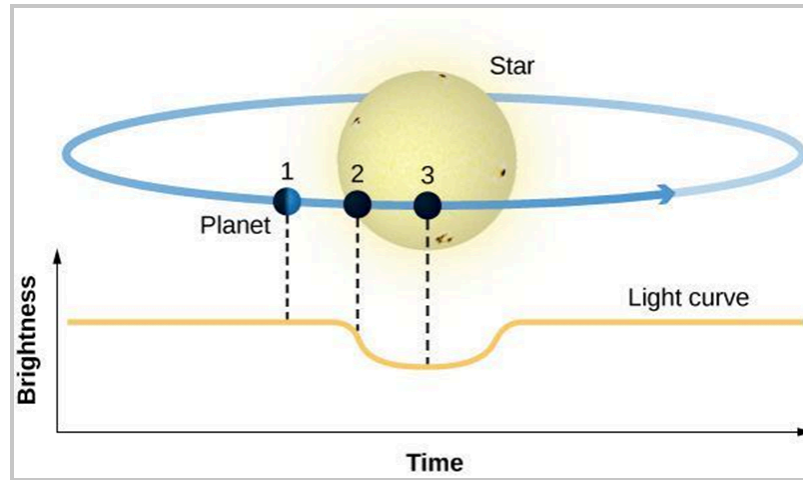


Figure 1. As the planet transits, it blocks some of the light from the star, causing a temporary dimming in the brightness of the star. The top figure shows three moments during the transit event and the bottom panel shows the corresponding light curve: (1) out of transit, (2) transit ingress, and (3) the full drop in brightness [13].

A shortcoming of this method is that some planets have orbits that last longer than the lifetime of a human such as in our Solar System Neptune takes 165 years to orbit [14]. It is impractical to detect certain exoplanets, such as COCONUTS-2b [15], with this technique. Another shortcoming of this method is the presence of variations of the host star's observed brightness caused by stellar activity and can mimic exoplanet transits. Thus, this must be accounted for when looking for exoplanets based on dips in the light curve. Additional limitations of the transit method are the low probability of properly aligned planet systems and the presence of astrophysical phenomena that may mimic transits and give rise to false detection positives [16].

Traditional methods used to distinguish true exoplanets from false positives include physical follow-up tests using other detection methods (which may also return a false positive) and statistical validation tools in a rigorous, multi-step process called vetting. The issue is that vetting takes a large amount of time and/or equipment to carry out [17].

Machine learning allows computers to learn, adapt, and improve from data. With respect to exoplanet detection, machine learning algorithms may be able to learn how to identify more minute differences of the observed data and thus reduce the possibility of overlooking exoplanets compared to traditional

methods. There are various methods for detecting exoplanets: Doppler measurements, transit observations, microlensing, astrometry, and direct imaging [18].

Machine learning is an emerging technology in the field of astronomy[19]. It has already demonstrated many benefits to this field. Consequently, machine learning has become an integral tool in contemporary astrophysics [20].

Other Methods of Exoplanet Detection

The radial velocity method is a newer method compared to the transit photometry method. The method, which shifts the wavelength of the star's light [21], involves using the doppler effect of light (commonly known as redshift and blueshift) to determine the acceleration of stars towards or away from us. This acceleration is caused by the gravitational force exerted on a star by the planet, causing the star to move in a relatively small circle of motion

In their study, Mayor & Queloz, claimed to have discovered a giant planet, 51 Pegasi, using the radial velocity method. This is the first exoplanet discovered to be orbiting around a main-sequence star [22].

Direct imaging, also called high-contrast imaging (HCI), stands out as the only method capable of directly observing giant gaseous exoplanets at wide separations, particularly in young systems [23]. Direct imaging consists of masking the light of the host star with a coronagraph or a nulling technique to reveal what surrounds it. It is a challenging technique, as the contrast, i.e., the brightness ratio between the planet and its host star, is of the order of

$$\frac{10^{-6}}{10^{-6}}$$

even in rather favorable cases [24].

Figure 2 shows the currently observed exoplanet population and which techniques were used to observe them.

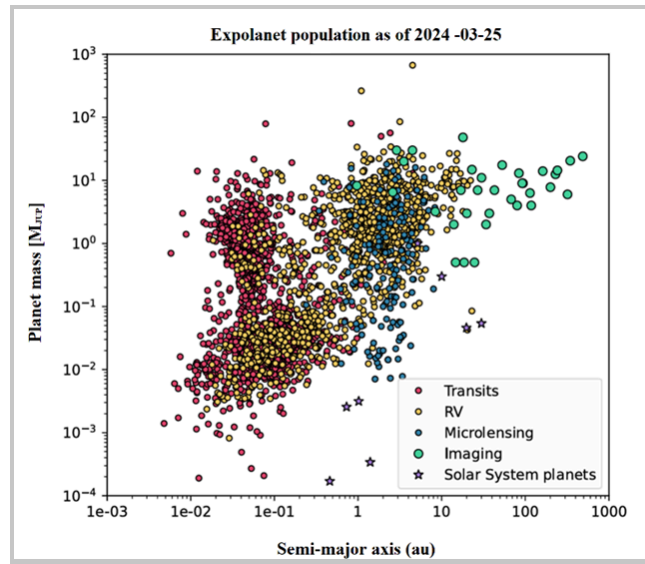


Figure 2. Proportion of exoplanets detected through various means. The most common ones are Transits and Radial-Velocity [21].

The transit method is used for this investigation due to its simplicity in the detection of transit dips. This means that it should be theoretically easier for the algorithms to distinguish a correct result, and they will thus gain a higher accuracy. Additionally, it is much easier to access swaths of data on transit photometry via NASA's Kepler mission data compared to any other method [26].

Understanding Transit Light Curves

The primary data product of the transit method is the light curve. Light curves are graphs that show the brightness of an object over a period of time [27].

When an object crosses between a star and the observer, some of the light is blocked by the object, slightly reducing the detected brightness. When this happens on regular intervals and occur for the same periods of time, astronomers can deduce that an object is orbiting the star. This can mean one of two things: the system is binary (two stars orbiting each other), or a planet has been detected. It is almost always the latter due to the other star also having their own brightness to make up for the loss [28].

Machine Learning in Astronomy

Machine learning is a branch of artificial intelligence that focuses on creating algorithms and models that allow computers to learn from data without being specifically programmed for a task [29].

Machine learning has been used to predict asteroid behaviors for hazard detection and has also been used for exoplanet classification in a seminal vein to this paper as it investigates machine learning in astronomy [30].

Examples of machine learning approaches are decision trees, neural networks, and logistic regression. In exoplanet research, automated classification systems were developed to support Kepler's candidate validation process. McCauliff employed random forests to classify Kepler candidates [31]. Shallue & Vanderburg used Neural networks in exoplanet identification, applying deep learning to Kepler light curves [32]. In exoplanet detection, labels typically distinguish confirmed planets from false positives. Thompson provided examples of validated Kepler classifications used for supervised learning tasks [33].

METHODOLOGY

Research Approach

Each model was trained, validated, and tested on the same data using the same input features: mean flux, standard deviation of flux, maximum flux and minimum flux. Any less features would reduce the expressivity of the machine learning models, potentially decreasing their effectiveness, any more and the models risk overfitting.

The Kepler Dataset labels each planet as either a confirmed exoplanet or a false positive. These labels were used as the ground-truth labels the models learned to predict and evaluate against within the test data set.

For this study, a list of Kepler Input Catalog (KIC) targets was prepared using the NASA Exoplanet Archive [34]. Confirmed planets and false positives were downloaded separately in .csv file format from the NASA Exoplanet Archive comprising 2,746 confirmed and 4,839 false positives. Each of these two lists was sorted in ascending order, and the first 50 KICs from each class were selected as the sample to perform the study, using stratified sampling to ensure both subsets retain balanced class distribution of the original dataset.

There are 50 confirmed exoplanets and 50 false positives extracted from the dataset that were extracted using the code from Annexure A, the output of which is seen in Annexure B. Using this as an input, the code from Annexure C is run to create the three machine learning models and create an output on their performances, which is shown in Annexure D.

The light-curve data for these KICs was accessed from the NASA Kepler archive Mikulski Archive for Space Telescopes (MAST) [35]. The time-series measurements for each KIC i.e., confirmed planets as

well as false positives, which constituted of time, flux, and flux uncertainty, were extracted using the Python programming language in Google Colab and downloaded in .csv files format. No additional features of light curves were obtained to keep the comparative analysis simple. The entire code snippet is provided on GitHub [50].

The Kepler dataset contained only 3 features: Time, flux, and the flux uncertainty. Time represents when each measurement was recorded. Flux is the observed radiant power per unit area and is used as a measure of brightness for a star in light curves. The flux uncertainty is what is used to represent stellar noise and systematic errors in the flux. These three sets of data are the components of a light curve, which is a measure of how the brightness of a star changes over a period of time with an element of uncertainty attached to it. They provide essential information for dealing with transit signals.

Each target in the dataset was assigned a binary class label indicating whether the observation corresponded to a confirmed exoplanet (assigned a label of 0) or a false positive (assigned a label of 1). This enables supervised machine learning models to make a decision boundary to discriminate between exoplanets and false positives. Binary classification is commonly used in exoplanet detection because each target must be classified as an exoplanet or not. McCauliff also used binary classification labels when developing automated machine learning classifiers for Kepler data[31].

There are many potential false positives, those being eclipsing binaries (a system with 2 stars orbiting each other), background eclipsing binaries, comets, stellar variability and instrumental artefacts, among others. These all have slightly different features that may be difficult for machine learning models to differentiate from, meaning some false positives will be harder to detect than others.

Data pre-processing

Before training the models, the dataset was normalized. The numerical features were scaled to zero mean and unit variance, as machine learning algorithms could be sensitive to differences in scale between input variables (models favor the features with greater scale more, even if the smaller scale variables are more predictive, and also leads to overfitting) [36].

Missing values present in the data set were handled using imputation techniques. Feature engineering was performed by deriving statistical features from the light curve time series, which are the mean, standard deviation, maximum, and minimum flux. These features were specifically engineered because they capture the key characteristics of the brightness data while keeping the model input dimensionality low. Four statistical features is the optimum number in the context of this research. Any number lower than four may lead to under-representation, while any more than four may lead to overfitting.

Finally, the dataset was labeled according to whether it was confirmed exoplanet or not (0 for confirmed planet and 1 for false positive). To avoid biases introduced in the training from class imbalance, the data

set was balanced for a 1:1 ratio of confirmed exoplanets to false positives. Balancing the classes in this way is important as models trained on highly imbalanced datasets can become biased towards the majority class.

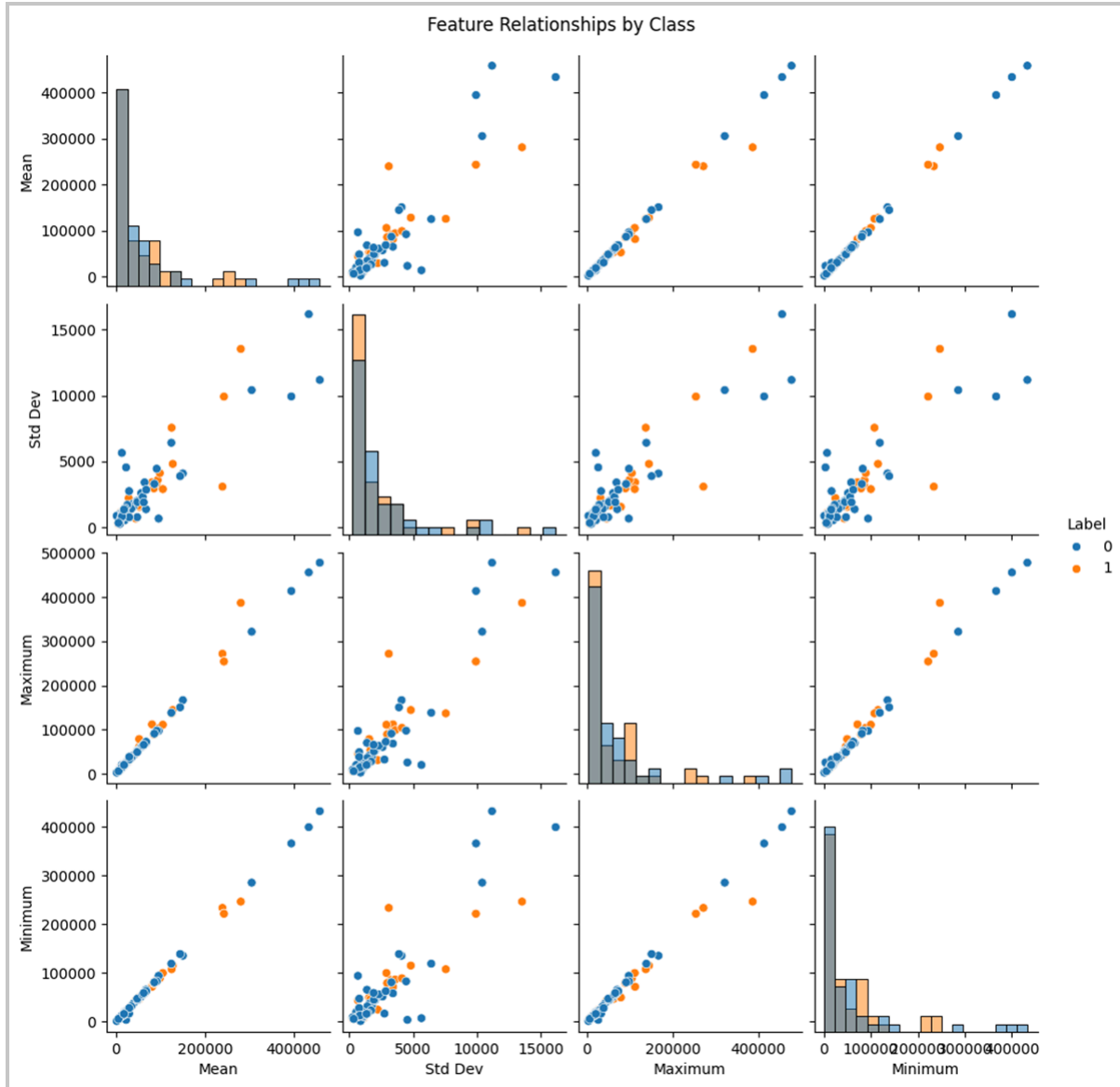


Figure 3. The various feature relationships used in training are separated by class where blue are negatives and orange are positives. The minimum, mean, standard deviation and maximum of the luminous flux of the light curves used to train the models are compared to each other in a set of 16 graphs.

Machine Learning Algorithms

Machine learning models such as logistic regression and decision trees are known for their strong interpretability and low computational requirements compared with deep learning and serve as excellent baseline performance indicators. The models employed in this research project are logistic regression, random forest, and shallow neural networks. Despite the limited scope, the results of the study are useful to determine what kinds of models will be effective for larger datasets. A benefit of this approach is that less time is needed for training and testing, and makes the models easier to interpret.

Logistic Regression

Logistic regression is the simplest classification model. It is a supervised machine learning model that predicts the probability that a given input belongs to a certain class, which in this case is whether the exoplanet is real. In this model, the input features are each multiplied by the weight of the edge and then added to the bias of its neuron. Then, the sigmoid function, the activation function, transforms the resulting values to a number between 0 and 1, reflecting the probability that the given input is an exoplanet [37].

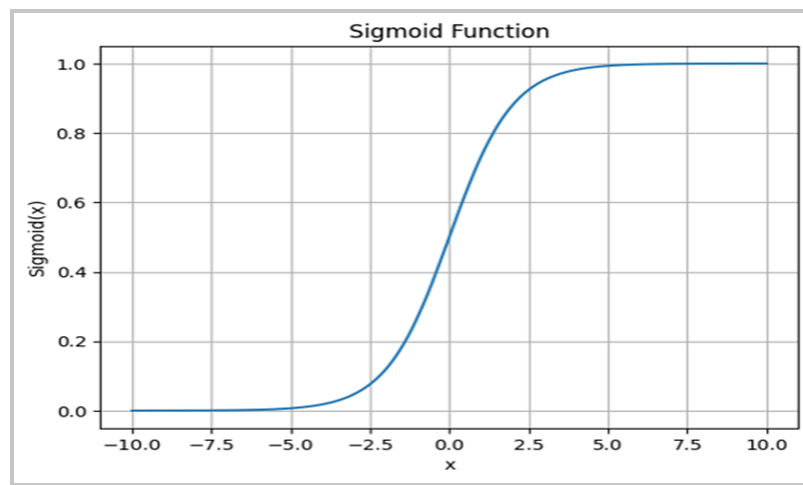


Figure 4. A graph of the sigmoid function, $\Phi(z)$. At $z = 0$, $\Phi(z) = 0.5$, and as z gets larger or smaller, $\Phi(z)$ gets closer to 1 and 0 respectively. In logistic regression, an output closer to 1 would represent a high confidence in the positive class, and one closer to 0 would represent a high confidence in the negative class. $\Phi(z)$ has a range of $[0,1]$ [36].

It has many benefits from being the simplest and easiest to implement. One of which is having higher computational speeds compared to other models as it uses the bare minimum number of neurons and edges. Another great benefit is that it is highly interpretable, so one can always deduce exactly what parts of the data being input are important. Because it does not contain any hidden layers, all the math can be

traced directly. However, this also means that it cannot handle very complex pattern recognition due to not having any hidden layers of neurons that could decipher them (called expressivity) [38]. It can only construct linear decision boundaries as it is a linear classifier (since the output of a logistic regression depends on the sum of the inputs and parameters) [39].

Random Forests

Decision trees

Decision trees are models that split the dataset based on features to create subsets of the original dataset. In other words, they are binary trees that recursively separate data via individual. If statements until the data that satisfies a certain set of parameters will always belong to a certain category (which in this case will be confirmed exoplanets and false positives) [40]. A decision tree is a directed graph where each internal vertex in the graph is a binary test; in this case, the tests are inequalities. If the test is true then one branch is taken, else the other branch is taken. Leaf vertices indicate classifications.

A strength of decision trees is their exceptional handling of non-linear patterns, allowing them to be more robust than other simple models. This is a great benefit if identifying data for exoplanets is not linearly separable. Another benefit of decision trees is easy interpretability, making it easier for scientists to extract what data is important in light curves [40].

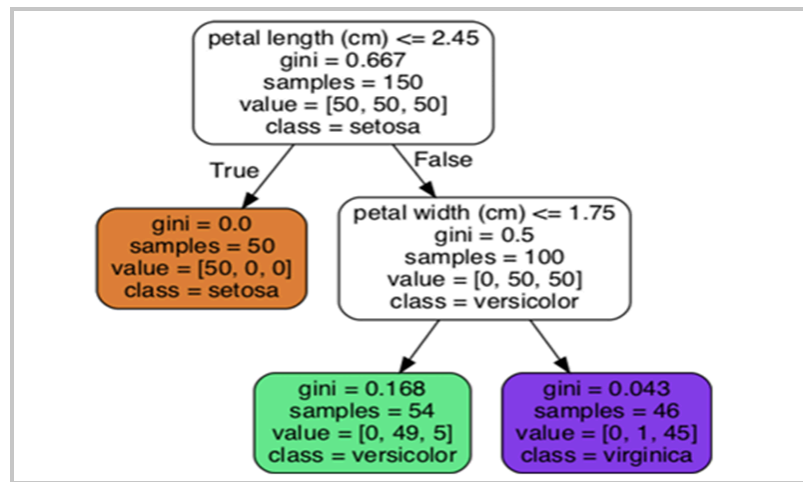


Figure 5. Decision tree example. This decision tree was trained from the iris dataset. The white boxes are where the dataset is split, and the colored boxes are the classes [40].

A major limitation of decision trees is that they are prone to overfitting. This can be mitigated using hyperparameter tuning (optimizing settings like minimum number of splits or maximum split depth), pruning (simplifying the tree by removing sub-optimal splits) and combining trees in an ensemble [40].

Another limitation of decision trees is that they struggle to model linear relationships due to them using step-like patterns which are unsuitable for smooth, continuous data [40].

Random forests

A random forest itself is an ensemble of decision trees. Each tree votes for the classification of an unknown object based on the input features, represented as a vector, that describes the object in question. The classification of the forest assigned to an object is the plurality vote of the random forest (the class with the most votes). The trees in random forest are constructed in such a way that attributes are randomly selected, so each tree makes a different set of splits [40]. McCauliff presents the application of machine learning techniques to the classification of exoplanet transit-like signals present in the Kepler light curve data for the first time and states that the random forest algorithm has been used previously to classify variable stars. This was the first time it had been used for exoplanet classification [31].

A random forest is a meta-estimator (a machine learning model that takes other models as inputs to modify, improve or combine behavior) that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [41]. Two techniques, bootstrap sampling and choosing a random subset of attributes at each split, break the correlation among the decision trees in the forest so that they approach the ideal of independent classifiers [42].

An important consideration to keep in mind is bias-variance tradeoff. Ensemble techniques aggregate results, so they reduce variance (overfitting) but may miss relations and increase bias (underfitting) [43].

Decision forests capture nonlinear relationships more effectively than linear models and often perform particularly well in astrophysical classification tasks. Random forests have been used for exoplanet classification by McCauliff [31].

Neural Networks

Neural networks share similarities to logistic regression models, except that there are hidden layers of neuron between the input layer and output layer that allows the model to have a greater ability to recognize more complex patterns in their inputs [44] and allows them to construct non-linear decision boundaries, thusly universal function approximators (meaning they can approximate any continuous function to arbitrary degree).

Neural network models consist of two things: neurons and edges. Neurons appear in layers, one for the output, and may contain any number of hidden layers between them, with the input layer being comprised of inputs instead. Each neuron in the input layer is connected to each neuron in the output layer by an edge, with each edge having a weight and its corresponding neuron having a bias (important to ensure the

decision boundary does not need to go through the origin). Each neuron has a certain activation function that takes the previous layer as inputs and returns a decimal value, called its activation. Specifically, the activation of each neuron gets multiplied by the weight, added to the bias and then normalized by a mathematical function (usually the sigmoid function or RELU (Rectified Linear Unit)) [32].

This causes neural networks to have more complex pattern recognition over other methods due to their additional layers. It's the reason they are almost always more accurate than other models. However, there is a risk of overfitting, where the increase of accuracy and precision does not correspond to the increase in computational power required to train and run the model, in which case it would be more effective to use a simpler model [45].

There are two types of neural networks: shallow neural networks (SNNs) and deep neural networks (DNNs). SNNs have up to one hidden layer of neurons and are thus relatively simple compared to DNNs, which have more than one. SNNs tend to be faster to train and run inference, which is why we will be using it here [46].

Training And Testing

For all machine learning algorithms, there are three phases: testing, validation and training. In the training phase, labelled data is fed into the algorithm, and an output is predicted. This is compared to what the output was supposed to be using a cost function. We use calculus to find the most important parameters to adjust to reduce the cost function for the next pass and repeat. Due to the small amount of data that can be extracted from the data set that is confirmed to be valid and does not contain any errors. The data will be split into a ratio of 6:2:2 of training to validation to testing to ensure that the model is trained effectively while also leaving ample data for testing the model's accuracy. Therefore, the dataset was split with 60% of the observations used for training, 20% used for validation and 20% used for testing.

Logistic regression used log-loss as its loss function and its optimizer is Limited-memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs). The strength of regularisation was optimized using manual grid search. Random forest used gini as its loss function, and the hyperparameters tuned were the number of estimators and the maximum depth. The shallow neural network also used log-loss as its loss function. The hyperparameters tuned were the size of the hidden layers, the activation function, the optimizer, and alpha (which controls L2 regularisation). All three had their hyperparameters tuned via manual grid search.

To test the generalizability of the results, a 7-fold cross-validation was performed where the data set was split seven different ways. The models were trained on the seven splittings. The results were reported using the mean and standard deviation of the evaluation metrics across the seven folds.

The random forest model had a maximum of 100 trees, with a maximum depth of 5. The Neural Network had 8 neurons in its hidden layer, for a total of 49 parameters.

The light curve data of targets was converted to statistical features of mean, standard deviation, maximum and minimum using Microsoft Excel and was then labelled. Annexure B provides the entire work performed in Microsoft Excel. Using Google Colab, Python was used for the machine learning algorithms and to generate predictions for the testing dataset. Using sklearn 1.9.0, the models produced predicted labels for each observation in the testing data, which were then compared with the actual class labels of the dataset.

Evaluation Metrics

The evaluation metrics used in this study were accuracy, precision, recall and F_1 . It is common to use these metrics in machine learning research as mentioned by Rainio [47].

Confusion matrix forms the basis of these metrics. Rainio [47] defines it as a matrix containing the counts of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) and are provided in the results section.

	Predicted positive	Predictive negative
Actual positive	True Positive (TP)	True Negative (TN)
Actual Negative	False positive (FP)	False Negative (FN)

Table 1. Outline of a confusion matrix

This is how confusion matrices, such as figure 6, are structured. These are used to present the statistics of the models.

These metrics are defined below. Accuracy is the overall classification of correctness. Accuracy is another measurement defined as the proportion of true instances retrieved, both positive and negative, among all instances retrieved [46].

The mathematical formula for accuracy (A) is shown in Equation 1 below,

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eq. (1)}$$

Precision (P) measures the number of correct instances retrieved divided by all retrieved instances. Precision is the ratio of correctly predicted planets to total planets.

The formula for precision is

$$P = \frac{TP}{TP + FP} \quad \text{Eq. (2)}$$

High precision means that the model produces few false positive predictions.

Recall (R) measures the number of the correct instances retrieved divided by all correct instances. Recall is the correctly detected planets out of actual planets,

$$R = \frac{TP}{TP + FN} \quad \text{Eq. (3)}$$

High recall means it predicts the positive outcome consistently

The F-score is defined as the weighted average of both precision and recall,

F-score usually means F_1 -score

F_1 score is the balance between precision and recall. Each model has a trade-off between precision and recall, so F_1 scores can compare how well the trade-off is made.

The formula for F_1 score is

$$F_1 = \frac{PR}{P+R} \quad \text{Eq. (4)}$$

A lower F_1 score means that the model is not as effective as one with a higher F_1 score.

The evaluation metrics were calculated separately for each seed. The mean and standard deviation of each metric were then calculated across the seven folds to show the average performance and stability of each model.

RESULTS

Model Performance Results

After completing the model training, the three machine learning algorithms were evaluated using the testing portion of the dataset. The purpose was to assess the performance of each model on their ability to correctly classify observations that were not seen during training.

The confusion matrices for each of the three models as calculated in Python programming language in Google Colab are reproduced below, while the entire code snippet is provided in Annexure C on GitHub [50].

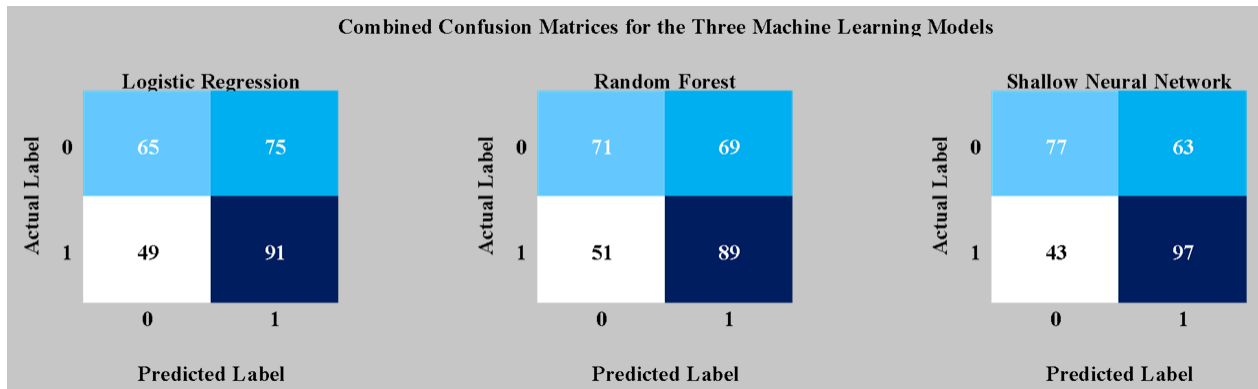


Figure 6. Confusion Matrices for the three machine learning models

Figure 6 shows the combined confusion matrix counts across the 14 random seeds. Since each test set contained 20 observations, each model made 280 predictions in total across the 14 seeds. Full details of confusion matrices are provided in Annexure D.

The combined confusion matrix counts help explain the model results. The shallow neural network identified the highest number of true positives and the lowest number of false negatives, meaning it was strongest at identifying actual confirmed planets. Random Forest identified the highest number of true negatives, meaning it was best at correctly detecting false positives. However, Random Forest also had the highest number of false negatives, meaning that it did not detect more confirmed planets.

The confusion matrices were further used in Python to calculate accuracy, precision, recall and F_1 score via Equations 1,2,3 and 4.

The confidence intervals for the accuracy, precision, and recall were computed using a Clopper–Pearson interval [48]. For the F_1 score, the Jaccard score [49] is identified as

$$J = \frac{TP}{TP + FP + FN} \quad \text{Eq. (5)}$$

Conditioned on $M = TP + FP + FN$. The count TP is binomial and thus the F_1 score is a monotonic transformation of the Jaccard score J :

$$F_1 = \frac{2J}{1+J} \quad \text{Eq. (6)}$$

Therefore, we compute the Clopper–Pearson interval of the Jaccard score identifying the number of successes as TP and the number of trials as M . This gives the confidence interval on the Jaccard score which is then transformed to the F_1 score using Eq. (6).

The results of this step are summarized below:

Model	Accuracy	Precision	Recall	F_1 Score
Logistic Regression	0.5571 ± 0.0375	0.5195 ± 0.0334	0.6500 ± 0.1214	0.5745 ± 0.0663
Random Forest	0.5714 ± 0.0554	0.5776 ± 0.0578	0.6357 ± 0.0679	0.5967 ± 0.0477
Shallow Neural Network	0.62141 ± 0.0393	0.6159 ± 0.0372	0.6929 ± 0.0822	0.6406 ± 0.0443

Table 2. The means and standard deviations of the accuracy, precision, recall, F_1 score given in Eqs. 1, 2, 3, and 4, respectively, for each model.

Table 2 shows that the shallow neural network achieved the highest mean accuracy, precision, recall and F_1 score across the seven random seeds, indicating that it is the best among the three classical machine learning algorithms in exoplanet detection. Logistic regression model acquired the second highest mean F_1 score while Random Forest achieved the lowest in it. Random Forest performed worse than the Logistic Regression model, so we can reasonably assume from this that the parameters provided in the training data is linear, as Random Forest struggles to model linear relationships that Logistic Regression is superior in. A clean threshold of 0.5 was used to determine the classes for all three models.

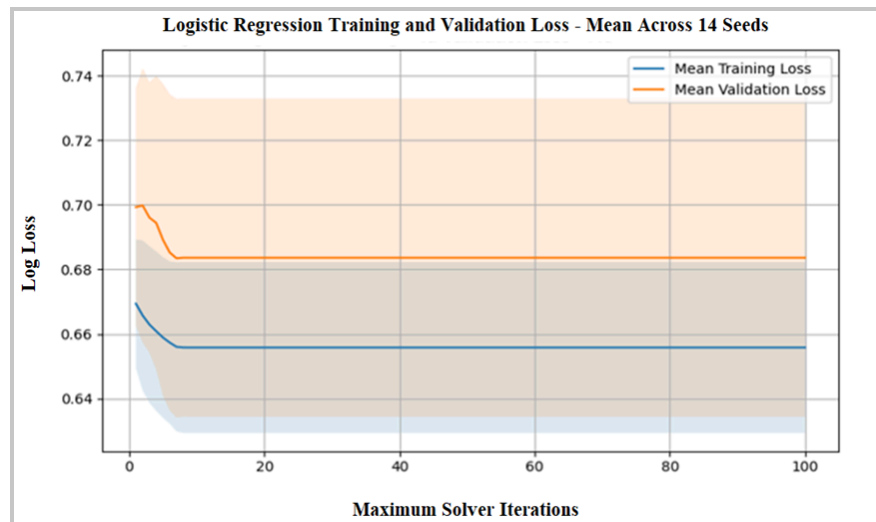


Figure 7. Logistic Regression Training vs Validation Loss

Mean training and validation log loss for Logistic Regression. Shaded regions show ± 1 standard deviation across the runs.

Figure 7 shows that Logistic Regression converged quite quickly, plateauing within the first few iterations. Training loss stayed slightly lower as the model was fit to the training data, but as the curves are mostly parallel to each other. Further training resulted in no improvement, proving the small limits that the model has.

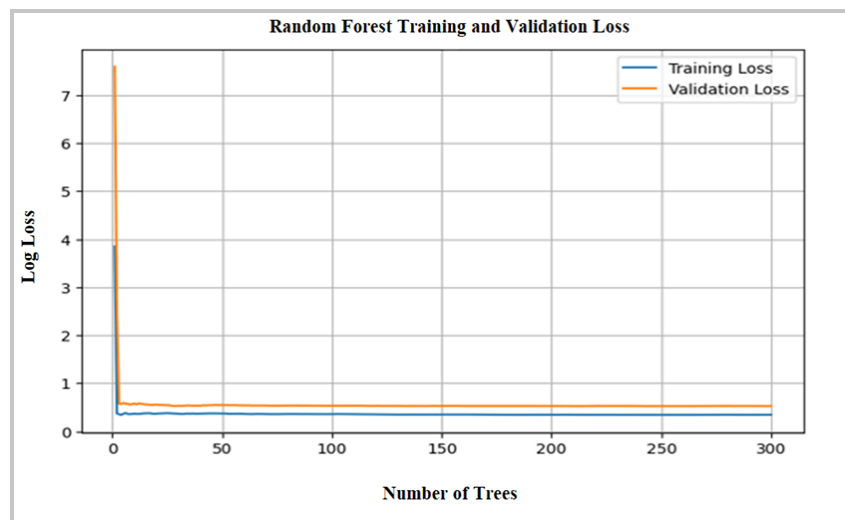


Figure 8. Random Forest Training vs Validation Loss

Mean training and validation log loss for Random Forest. Shaded regions show ± 1 standard deviation. Random forest loss dropped sharply as more trees were added and stayed stable. Validation loss remained higher but stayed parallel with training loss as more trees were added. Adding more trees still kept both curves stable, so the random forest did not succumb to overfitting.

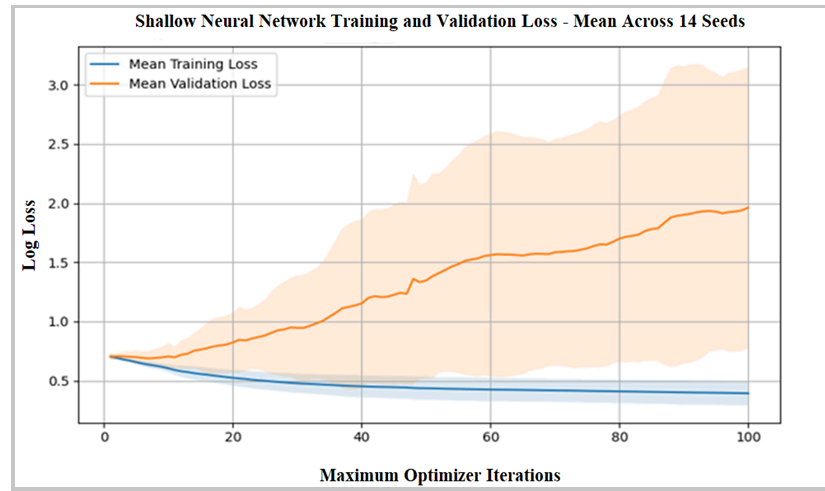


Figure 9. Shallow Neural Network Training vs Validation Loss

Mean training and validation log loss for the Shallow Neural Network. Shaded regions show ± 1 standard deviation.

Despite being the superior model of the three in terms of results, the shallow neural network showed clear signs of overfitting. Validation loss rose quite steeply compared to the gradual fall of training loss, meaning that the model started to memorize the training dataset. The wide variations in validation loss also show that the SNN was sensitive to changes in how the dataset was split.

In figure 9, the orange line diverges from the blue one. This is a good indication of overfitting, justifying the use of only 4 features as any more would worsen the model's performance. Additionally, due to the small dataset available, using more features would cause a higher chance for memorization of the training dataset rather than actually learning the trends.

Key Observations

Following are key observations from the results: Shallow neural network achieved the highest mean F_1 score, suggesting that it provided the most balanced overall performance among the three models. On the other hand, Logistic regression achieved the lowest mean F_1 score. Although it is the simplest model, its

relatively high recall suggests that it could distinguish between confirmed planets and false positives with only four statistical features.

Also, The Shallow Neural Network's standard deviation for Recall is abnormally large compared to its other standard deviations, and the other standard deviations for Recall are all above 10%. This leads to the theory that detecting exoplanets correctly may be a bit too complex with the provided parameters for these three models.

Even with limited statistics, these models can still contribute to astrophysics and identify patterns in very little time. These findings supports the growing role of machine learning in astronomy. It also demonstrates the value of comparing multiple machine learning algorithms when investigating exoplanet detection problems.

DISCUSSION

Interpretation of Results

Logistic regression performs well on problems when there is an approximately linear relationship between features and the target variables, leading us to believe, due to its performance, that the inputs do not separate the light curves in a linear fashion. This lines up with the variability of stellar brightness with its non-uniform patterns. The shallow neural network model has the highest mean F_1 score than the logistic regression model due to it having additional hidden layers that allow for more flexibility in the model compared to logistic regression which has none.

The results also support the potential use of established exoplanet vetting methods. The Kepler DR25 Robovetter [51] showed that its reliability varied considerably under differing conditions, showing that the effectiveness of these methods on the characteristics of what is being investigated, In this study, all three models were able to distinguish confirmed planets and false positives only using 4 basic features. Direct comparison between these is currently not possible due to them not being evaluated using the same sample using comparable measures. Nonetheless, machine learning models can still extract useful information from the light curves. This suggests that machine learning could provide a complementary method of automated candidate classification and, with richer features and larger training datasets, could potentially improve upon existing vetting approaches in some operating conditions.

Limitations

While the study provides meaningful insights, several factors explain the limitations. They are explained below:

1. There may be some bias in the data which is being analyzed due to observation bias.

2. Additionally, the features used are very simple and few, meaning these features may not capture the full complexity of exoplanet transit signals. Models that can capture the equivariant nature of time series information (that is, how shifting the actual times do not affect the validity of a transit) is a solution to this. Model bias should also be considered. Efforts were made to reduce this through consistent data preprocessing, but some degree of bias will always be prevalent in any machine learning investigation.
3. No details about the nature of false positives is available from the Kepler dataset, so it is not possible to determine what kinds of false positives give the models the most difficulty.
4. The 4 features used are simple, which may be limiting the expressivity of the models.
5. A threshold of 0.5 was used universally, so the models may not be performing optimally. A greater threshold means less false positives, but more false negatives and vice versa, which may lead to greater performance.

Further research with the use of larger datasets and more complex features may better evaluate model performance in exoplanet detection tasks.

CONCLUSION

Following the empirical results obtained from the dataset of 100 observations, it can be concluded that the overall shallow neural network performed best, as evidenced by its higher mean F_1 score compared to the other two models.

It achieved the highest mean precision and the highest mean F_1 score across the seeds. It also achieved the highest mean recall, making it the strongest model overall for this dataset and selected feature set.

Although this investigation did not directly benchmark the three models against traditional methods under identical conditions, the results show that machine learning can provide meaningful classification even with limited data. It can be seen that established methods have performance limitations under certain conditions. The present finding support that machine learning models can be used at the very least as a complementary approach to traditional methods. With larger datasets and more informative features, models may be capable of improving exoplanet classification even further, including in scenarios that other methods are less reliable. Demonstrating any superiority would involve using the same observations and criteria, which are outside of the scope of the paper.

Overall, these findings demonstrate that model choice affects classification performance and that more flexible models tend to perform better when dealing with complex data patterns.

REFERENCES

1. Perryman M. (2018). "The Exoplanet Handbook":
<https://onlinelibrary.wiley.com/doi/full/10.1111/maps.13308> [ISBN #978-1-108-41977-2]
2. Malik, A., et al. (2022). "Exoplanet detection using machine learning.":
<https://doi.org/10.1093/mnras/stab3692>
3. Venturini et al. (2024). "A fading radius valley towards M dwarfs, a persistent density valley across stellar types": <https://doi.org/10.1051/0004-6361/202349088>
4. Mayor and Queloz. (1995). "A Jupiter-mass companion to a solar-type star":
https://ui.adsabs.harvard.edu/link_gateway/1995Natur.378..355M/doi:10.1038/378355a0
5. Cenadelli, D., Bernagozzi, A. (2015). "Youth plus experience: the discovery of 51 Pegasi b":
<https://doi.org/10.1140/epjh/e2015-60041-5>
6. Barbara E. McArthur et al. (2004). "Detection of a Neptune-Mass Planet in the ρ 1 Cancri System Using the Hobby-Eberly Telescope": <https://iopscience.iop.org/article/10.1086/425561#artAbst>
7. Chris Impey (2023). "Worlds Without End: Exoplanets, Habitability, and the Future of Humanity": <https://doi.org/10.7551/mitpress/14361.001.0001>
8. Ayan bin Rafaih, Zachary Murray (2025). "Detecting False Positives With Derived Planetary Parameters: Experimenting with the KEPLER Dataset":
<https://doi.org/10.48550/arXiv.2508.13801>
9. Borucki, W. J., et al. (2010). "Kepler Planet- Detection Mission: Introduction and First Results":
<https://www.science.org/doi/abs/10.1126/science.1185402>
10. George Ricker et al. (2014). "Transiting Exoplanet Survey Satellite":
<https://doi.org/10.1117/1.JATIS.1.1.014003>
11. Jean Creighton. (2019). "The Survey of astronomy":
<https://wisconsin.pressbooks.pub/astronomy/chapter/chapter-21-section-21-5-exoplanets-everywhere-what-we-are-learning>.
12. Kennedy Jnr Izuagbe (2015). "The Transit Method for Exoplanet Detection":
10.13140/RG.2.1.1297.0323
13. Jean Creighton (2019). "Planets Beyond the Solar System: Search and Discovery.":
<https://wisconsin.pressbooks.pub/astronomy/>
14. <https://lco.global/spacebook/solar-system/neptune/14>
15. Zhang, Z. et al. (2021). "The Second Discovery from the COol Companions ON Ultrawide orbiTS (COCONUTS) Program: A Cold Wide-Orbit Exoplanet around a Young Field M Dwarf at 10.9 pc": <https://doi.org/10.48550/arXiv.2107.02805>

16. Deeg & Alonso. (2018). "Transit Photometry as an Exoplanet Discovery Method":
<https://arxiv.org/abs/1803.07867>
17. Sarah Huang, Chen Jiang. (2025). "Machine Learning for Exoplanet Discovery: Validating TESS Candidates and Identifying Planets in the Habitable Zone": <https://arxiv.org/html/2512.00967v1>
18. Fischer et al. (2015) "Exoplanet Detection Techniques":
<https://doi.org/10.48550/arXiv.1505.06869>
19. Dalya Baron. (2019). "Machine Learning in Astronomy: A Practical Overview":
<https://arxiv.org/pdf/1904.07248>
20. Reihaneh Kairimi et al. (2025). "Machine Learning for Exoplanet Detection: A Comparative Analysis Using Kepler Data": <https://doi.org/10.48550/arXiv.2508.09689>
21. Zurlo, A. (2024). "Direct imaging of exoplanets": <https://arxiv.org/abs/2404.05797>
22. T. Guillet et al. (1995). "Giant Planets at Small Orbital Distances":
<http://doi.org/10.48550/arXiv.astro-ph/9511109>
23. Katherine B. Follette. (2023). "An Introduction to High Contrast Differential Imaging of Exoplanets and Disks": <https://doi.org/10.1088/1538-3873/aceb31>
24. Leaf Swordy et.al. (2025). "The Dispersion Leverage Coronagraph (DLC): A nulling coronagraph for use on primary objective grating telescopes."
<https://doi.org/10.48550/arXiv.2507.11745>
25. G. Ingrosso et al. (2014). "Measuring polarization in microlensing events":
https://ui.adsabs.harvard.edu/link_gateway/2015MNRAS.446.1090I/doi:10.48550/arXiv.1410.3594
26. Joshua N. Winn. (2014). "Transits and Occultations": <https://arxiv.org/pdf/1001.2010>
27. F. Dououdi et al. (2019). "Light Curve Analysis of Ground-Based Data from Exoplanets Transit Database": DOI:10.48550/arXiv.1910.11438
28. Chetan Abhijnanam Bora et al. (2024). "Temporal trends in asteroid behaviour: a machine learning and N-body integration approach":
<https://academic.oup.com/mnras/article/534/1/415/7750043>
29. Tom M.Mitchell. (1997). "Machine learning":
<https://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf> [ISBN: 0070428077]
30. Ibrahim M.Khalil et.al. (2023). "Detecting Asteroids and Comets using Machine Learning and Deep Learning": 10.21608/MSAENG.2023.291924
31. McCauliff, S. et al. (2014). "Automatic Classification of Kepler TCEs":
<https://arxiv.org/abs/1408.1496>
32. Shallue & Vanderburg (2018). "Identifying Exoplanets with Deep Learning: A Five Planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90":
<https://arxiv.org/pdf/1712.05044>
33. Thompson et al. (2018). "A Machine Learning Technique to Identify Transit-Shaped Signals":
<https://arxiv.org/abs/1509.00041>
34. NASA Exoplanet Archive (n.d.). Kepler KOI Documentation:
https://exoplanetarchive.ipac.caltech.edu/docs/Kepler_KOI_docs.html

35. Mikulski Archive for Space Telescopes (MAST) (n.d.). MAST Portal:
<https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html>
36. João Manoel Herrara Pinheiro. (2025). “The Impact of Feature Scaling In Machine Learning: Effects on Regression and Classification Tasks”: <https://doi.org/10.48550/arXiv.2506.08274>
37. Edvaldo Domingos et al. (2021). “Experimental Analysis of Hyperparameters for Deep Learning-Based Churn Prediction in the Banking Sector”
<https://doi.org/10.3390/computation9030034>.
38. Minh Tri Lê. (2026). “Efficient Neural Networks for Tiny Machine Learning: A Comprehensive Review”: <https://dl.acm.org/doi/10.1145/3798276>
39. Moo K. Chung. (2020). “Introduction to Logistic Regression”:
<https://doi.org/10.48550/arXiv.2008.13567>
40. Aurélien Géron. (2019). “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd Edition)” [ISBN: 978-1492032649]
41. Salman H.Ahmad. (2024). “Random Forest Algorithm Overview”:
<https://doi.org/10.58496/BJML/2024/007>
42. Michał Iwaniuk et al. (2025). “The Impact of Bootstrap Sampling Rate on Random Forest Performance in Regression Tasks”: <https://doi.org/10.48550/arXiv.2511.13952>
43. Haixiang Lan et al. (2025). “The Bias-Variance Tradeoff in Data-Driven Optimization: A Local Misspecification Perspective”: <https://doi.org/10.48550/arXiv.2510.18215>
44. Michael A. Nielson. (2015). “Neural Networks and Deep Learning”: Determination press.
<http://neuralnetworksanddeeplearning.com/chap1.html>
45. Fazel Hesar & Foing. (2024). “Evaluating Classification Algorithms for Exoplanet Detection”:
<https://doi.org/10.48550/arXiv.2402.15874>
46. Introduction to Machine Learning course “Classification: Accuracy, recall, precision, and related metrics”:
<https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>
47. Rainio et al. (2024). “Evaluation metrics and statistical tests for machine learning”:
<https://doi.org/10.1038/s41598-024-56706-x>
48. H. Dalianis. (2018). “Clinical Text Mining”: https://doi.org/10.1007/978-3-319-78503-5_6
49. Gonzalo Travieso et al. (2024). “An Analytical Approach to the Jaccard Similarity Index”:
<https://doi.org/10.48550/arXiv.2410.16436>
50. GitHub repository: <https://github.com/Arsal-Abbas-dev/arsal-research-paper>
51. Thompson et al. (2018). “Planetary Candidates Observed by Kepler. VIII. A Fully Automated Catalog With Measured Completeness and Reliability Based on Data Release 25”:
<https://doi.org/10.48550/arXiv.1710.06758>

Annexure A

Annexure A is the code used to download the data from the dataset. It is located in the GitHub repository [50].

Annexure B

Kep ID	Mean	Std Dev	Maximum	Minimum	Label
KIC-2571238	239667.02	3087.79	271941.97	233762.08	1
KIC-2444412	127959.20	4816.68	144642.75	115028.22	1
KIC-2441495	52543.13	1761.86	61829.45	46378.57	1
KIC-2692377	91727.51	3190.13	103294.02	85606.36	1
KIC-2557816	16980.65	669.90	17921.30	15651.01	1
KIC-2713049	13793.46	420.75	14912.54	12951.91	1
KIC-757450	10616.23	1113.33	15399.27	7479.82	1
KIC-1026957	125325.94	7562.99	137161.73	107483.95	1
KIC-1161345	280935.05	13544.83	387066.75	246690.39	1
KIC-2302548	52444.17	1551.56	78951.80	49422.92	1
KIC-1432789	11333.52	506.30	12572.28	10392.06	1
KIC-1718189	27716.46	1087.93	29280.64	23879.52	1
KIC-2165002	9029.05	364.69	9596.97	7934.60	1
KIC-1871056	81303.00	3418.74	112174.60	71416.95	1
KIC-1865042	50010.65	1643.63	53989.57	45614.66	1
KIC-1725016	10450.08	682.71	11360.85	9150.62	1
KIC-2716853	6927.09	245.16	7342.88	6462.81	1
KIC-2715135	21535.72	1101.22	23173.94	19321.01	1
KIC-2164169	16969.67	739.76	17799.97	15306.14	1
KIC-2574338	9717.54	359.11	10440.65	8995.31	1
KIC-2584163	11860.38	422.42	12382.25	10963.22	1
KIC-2721030	7981.08	377.82	8799.86	7333.27	1
KIC-2693736	12130.46	547.25	13106.45	11054.07	1
KIC-2304320	44468.02	671.14	45813.60	43246.16	1
KIC-2307415	85484.35	2959.66	89720.51	78947.25	1
KIC-2161536	7595.33	379.16	8455.56	6827.42	1
KIC-2556650	5438.48	341.74	6082.11	4772.94	1
KIC-2439243	5883.81	292.50	6516.03	5381.29	1
KIC-1872821	14514.22	1152.18	16059.81	11762.90	1
KIC-2142522	93981.47	3587.79	98936.76	86243.03	1

*Comparative Investigation of Machine Learning Model Performances for Exoplanet Transit Detection
Using Kepler Data*

KIC-2718778	7156.89	286.49	7521.83	6575.21	1
KIC-1996180	35185.23	1787.86	37815.92	32114.78	1
KIC-1849702	28914.21	2218.21	31507.43	24372.55	1
KIC-1995519	5646.37	285.61	6117.47	5056.55	1
KIC-2442448	25012.07	980.23	26294.13	23218.75	1
KIC-2716801	6837.78	251.63	7178.21	6275.88	1
KIC-2715695	5648.22	277.58	6014.02	5144.87	1
KIC-1717722	6819.43	366.37	7684.91	6333.63	1
KIC-2717158	11942.93	592.85	12901.51	10720.59	1
KIC-2581316	243057.67	9925.56	254479.77	221717.61	1
KIC-2581554	98777.22	4115.59	104542.52	88912.83	1
KIC-1718958	105616.93	2901.29	111432.13	99778.74	1
KIC-1724719	34885.87	1528.19	37557.32	31967.09	1
KIC-1873513	6805.25	488.94	7313.12	6002.13	1
KIC-2831055	42690.17	1841.33	45191.81	39417.23	1
KIC-2832589	9237.30	293.40	10214.00	8668.36	1
KIC-2837111	16349.32	1085.58	18076.39	14420.51	1
KIC-2849805	66783.91	1386.25	80439.04	64069.73	1
KIC-2853093	9258.51	397.47	73596.18	8514.08	1
KIC-2854181	10338.60	315.12	10797.78	9621.96	1
KIC-1996679	36629.20	1764.57	38951.29	32954.68	0
KIC-1571511	57000.45	2599.69	60879.19	51332.20	0
KIC-1433962	8790.50	523.67	9673.18	7776.00	0
KIC-2157247	24157.36	1326.54	26191.91	21304.21	0
KIC-1432214	7584.49	306.24	9907.95	6148.84	0
KIC-1722276	19033.34	1129.72	20337.42	16789.15	0
KIC-892772	11782.71	904.88	13076.31	10109.59	0
KIC-1027438	47747.29	2007.53	52265.92	43932.63	0
KIC-1996399	16852.67	759.82	17975.47	14870.43	0
KIC-2166206	64599.24	3410.28	68801.91	57930.89	0
KIC-2021440	9742.64	1038.41	11333.32	7981.77	0
KIC-2014991	150639.79	4090.65	167000.31	135315.55	0
KIC-2141783	39128.25	1559.01	41700.05	36588.46	0
KIC-1725415	21740.76	916.08	23278.03	20143.56	0
KIC-2142501	144487.10	3886.19	150850.64	138780.12	0
KIC-1995732	7324.74	503.28	8171.55	5851.08	0
KIC-2019076	6951.20	435.54	7624.47	5555.41	0
KIC-1575690	7582.83	547.19	8486.10	6293.46	0

KIC-2167890	96126.08	664.72	97456.77	93924.73	0
KIC-2166200	1876.06	872.11	3228.15	775.31	0
KIC-1433980	14053.58	1107.17	15462.06	10730.16	0
KIC-2010191	13425.58	5649.89	20435.34	6717.98	0
KIC-1293031	91783.70	4448.62	97994.23	82522.70	0
KIC-1724513	48494.51	1922.73	51255.39	44744.73	0
KIC-1575873	433737.02	16182.82	455850.10	399973.10	0
KIC-1430305	67782.58	1375.50	70422.53	65285.07	0
KIC-1161949	34473.00	1401.77	36502.84	31053.90	0
KIC-1297263	48121.99	772.68	49538.72	46916.36	0
KIC-1572948	19897.78	534.45	20504.96	19327.95	0
KIC-1725193	25725.22	1704.51	28203.74	22719.00	0
KIC-1870398	17458.86	1163.81	19632.73	14730.60	0
KIC-2010607	394524.31	9935.43	413862.28	366420.75	0
KIC-2015013	60408.24	2307.96	64042.67	55754.60	0
KIC-2163226	11923.02	684.87	12716.81	10618.15	0
KIC-2165352	21664.83	1034.40	23396.25	19478.36	0
KIC-2161623	22961.98	4547.88	26130.10	3232.54	0
KIC-1026032	16497.27	1047.46	18259.53	13242.56	0
KIC-1433531	68143.13	2851.23	72871.52	62352.22	0
KIC-1717528	20165.08	1444.18	22562.79	16966.80	0
KIC-2012722	305386.84	10418.66	321795.16	285719.90	0
KIC-2013883	458479.12	11189.67	477897.44	432857.50	0
KIC-2162994	29714.43	2748.80	32711.57	16164.57	0
KIC-2167444	30058.36	766.28	38777.51	27572.28	0
KIC-1293046	9008.44	259.49	9596.00	8582.32	0
KIC-1434395	124707.21	6419.98	138574.78	118919.59	0
KIC-1868404	86547.06	3293.11	91349.88	80604.09	0
KIC-1995712	13837.27	827.44	15161.93	12423.25	0
KIC-2140389	5846.63	332.76	6391.38	5196.35	0
KIC-2302758	18292.47	1344.62	20414.58	15583.13	0
KIC-2010152	62669.97	1896.79	66049.88	58681.97	0

Annexure C

Annexure C is the code used to run the machine learning models. It is located in the GitHub repository [50].

