**A Machine Learning Approach for the Identification of the Habitable Planets using the Astronomical Data**

Uzair Ahmed

Hassan Ahmed

**Abstract**

We have a growing need to manage terabytes of astronomical data produced every night. A huge amount of data is available on exoplanets that needs to be processed and classified. Machine learning is the modern solution to classification problems in large data sets, with specialized models to carry out the task automatically with high accuracy. We assessed the potential of neural networks and decision trees as habitability classifiers. We worked on habitable exoplanet catalog, prepared a dataset suitable for machine learning and split it 7:3 in training and testing sets. We also applied the latest bagging and boosting methods to decision trees to optimize learning and prevent overfitting. We found XG boost to provide the most accurate and precise predictive model. Moreover, using XG boost We found planet’s radius and solar insolation to be the most significant factors when determining habitability of an exoplanet.

# 

# CONTENTS

# 

# 1 Introduction……..….………………...…..…..3

# 2 Methods…………….…………….…..…........4

# 3 Results…………..…..….……….…….……...7

# 4 Discussion…..……...………….………….…11

# 5 References…..…………………………….…13

# 

# 

# 

# 

# 

# 

# 

# 1 INTRODUCTION

In the recent years we have observed a surge in the amount of astronomical data available to us via telescopes and observatories [1] (see Figure 1). This presents us with the problem of analyzing terabytes of data that pushes our computational power to its limits. This creates a huge gap between the technological advancements and our ability to manage the data they produce. Fortunately, recent developments in machine learning techniques have given us a way to deal with these large data sets. One of the areas in astronomy where it can be applied effectively is in our search for exoplanets, where telescopes like Kepler and TESS, and tens of observatories around the world, provide us with mountains of data about foreign star systems [2].

Most of machine learning efforts so far have been directed towards automatically identifying the false positives among the detected exoplanets [3]. However only limited research has yet been conducted in the use of machine learning for classification of these exoplanets. Some work has been done on Habitable Exoplanet Catalogue [4], to which existing machine leaning techniques have been applied [5], and the accuracy of different supervised learning algorithms has been tested [6]. This provides us with an opportunity to dig deeper into this area and test the effectiveness of latest machine learning models on existing data. This research will greatly highlight the potential of machine learning models as habitability classifiers and provide us with useful insights as to how different parameters can influence the classification of a planet. Hopefully this will make way for new astronomical research and discoveries.

Chart, bar chart

Description automatically generated

*Figure 1: Evolution of data available via telescopes over time [1**]*

# 2 METHOD

**2.1 Data**

Every machine learning project follows certain definite steps (Figure 2). It all starts with the original data set, which I obtained from planetary habitability laboratory (PHL) [4]. They have a complete catalogue of confirmed exoplanets extensively analyzed and classified according to habitability. This dataset had 4049 datapoints (exoplanets) and 55 of them were classified as potentially habitable. The dataset presented 86 planetary and stellar features collected from various sources.

![Diagram

Description automatically generated]()

*Figure 2: How a machine learning model is built.*

# After collecting the necessary data comes the part of preparing appropriate data to be used by the machine learning model. This is the most important part of the project because the model is only as good as the data it learns from. The habitability of a planet is centered around the presence of liquid water on its surface, the “Handbook of exoplanets” [7] provides a comprehensive list of astronomical parameters that influence this (see Figure 3). Of these only some are directly observable (blue) and while others are derived (green) or theoretically modeled (orange). Unfortunately, all these features were not available in PHL’s catalog, and even the ones available had missing values for several datapoints. Using the normalization method to fill in the missing values (as usually done in machine learning projects) resulted in more than 50% being made up. Realizing that training on this data will not result in a reliable predictive model, we decided to reduce the features to ones which were readily available for most exoplanets (see Table 1). We also wrote a python program that would run though the data set and eliminate datapoints missing any such features. At the end we were left with 2913 datapoints with all real data. [8]

|  |  |  |
| --- | --- | --- |
| **Planetary features:** | **Stellar features:** | **Habitability features:** |
| Mass (Earth Masses) | Mass (Solar Units) | Hab-zone inner radius (AU) |
| Radius (Earth Radii) | Radius (Solar Units) | Hab-zone outer radius (AU) |
| Period (Days) | Effective Temperature (K) | Habitability flag (0 or 1) |
| Semi-Major Axis (AU) | Star Luminosity (Solar units) |  |
| Equilibrium Temperature (K) |  |  |
| Mean Insolation (Earth Units) |  |  |

*Table 1: Astronomical features chosen for the project*

**![Chart, timeline, bubble chart

Description automatically generated]()**

*Figure 3: Features that affect the presence of liquid water on the surface of a planet [7]*

30% of this dataset was reserved for final testing to determine the accuracy of the ML model. The rest of 70% was used to train the models. This 7:3 division of the dataset was found to give optimal results. The next step in the process was to build ML models suitable for the task of classification.

**2.2 Machine learning models**

In data science connected neural network and decision trees (often used with optimization algorithms e.g. bagging and boosting methods that we used) are well known for their effectiveness in classification problems, so we decided to use them for habitability classification. Below are the technical details of the methods in use.

**2.2.1 Connected neural network**

The main packages we used were keras and scikit-learn. In order to bypass any initialization complications, we used the preprocessing package. In splitting the data, 30% was divided equally into the validation and test data, and the rest 70% was used for training. The following architecture was employed using fully connected layers (i.e. Dense): 3 hidden layers (100 layers, ReLU activation) and 1 output layer (1-layer, sigmoid activation). We first used SGD as the optimizer, but the loss went down much slower (gradual drop as opposed to a steeper drop) than when Adam was used - and hence Adam made the cut for the final model. To reduce overfitting to a minimum, we employed L2 regularization and used dropout with a probability of 0.3 (which fared better than a probability of 0.25), based on experimental results. We did not use the early stopping method because L2 regularization gave a better loss slope and did not have the tell-tale overfitting U-shape at any number of epochs (cycles). The batch size was a standard 32. Again, due to the nature of the data, we used a relatively simple neural network architecture with the main, oft-successful techniques instead of employing too many methods.[9]

**2.2.2 Random forest classifier**

Random forest is a supervised learning algorithm which is used for both classification as well as regression. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction. Random forest is one of the best examples of the bagging optimization of the machine learning models.

This method is good because a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The reason for this wonderful effect is that the trees protect each other from their individual errors. While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. [10]

**2.2.3 Ada boost**

Boosting is a way for weak learners to be transformed into strong learners. Each new tree is a fit for a revamped version of the original data set when boosting. AdaBoost is one of the first and one of the highest performing boosting algorithms to be adapted in solving practices.

A single classifier may not be able to accurately predict the class of an object, but when we group multiple weak classifiers with each one progressively learning from the others' wrongly classified objects, we can build one such strong model. The basic classifier in our case was a simple decision tree. In each iteration it improves by adding more weight to the wrongly classified samples so that they are classified correctly in the next time. The process of the creation of the decision stumps and the weight assignment continues until either all points have been correctly classified (which practically is impossible) or maximum level of iteration is reached. [11]

**2.2.4 XG boost**

Developed in 2016 XG Boost (eXtreme Gradient Boosting) is an optimized gradient boosting algorithm using parallel processing, tree-pruning, handling missing values and regularization to avoid bias. XG boost improves upon the base GBM (Gradient Boosting Machines) framework through systems optimization and algorithmic enhancements.

XG Boost takes the best parts from the boosting and the bagging method and adds upon them some additional features i.e. the sequential tree growing, the minimization of the loss function using while using the gradient descent, the parallel processing, and the regularization of the parameters.

XG Boost was developed to increase speed and performance, while introducing regularization parameters to reduce overfitting. In it several weak learners, use regression trees (CART) for the sequential learning process. For each iteration (i) which grows a tree (t), scores (w) are calculated which predict a certain outcome (y). The learning process aims to minimize the overall score which is composed of the loss function at i-1 and the new tree structure of t. This allows the algorithm to sequentially grow the trees and learn from previous iterations. Gradient descent is then used to compute the optimal values for each leaf and the overall score of tree t. XG Boost dominates structured or tabular datasets on classification and regression predictive modeling problems. [12]

# 3 RESULTS

The accuracy of machine learning models in classification problems is determined using a confusion matrix (see Figure 4 & Eq.1). The results for each of the models and their accuracy is presented in this section. For methods using decision trees namely random forest, ada boost, xg boost, we were also able to extract feature importance graphs representing how much weightage was given to each of the data features while determining habitability. Note that through out the report, class = 0 represents non habitable planets, and class = 1 represents habitable planets.

Graphical user interface, text, application, table

Description automatically generated

*Figure 4: Confusion matrix*

[Eq.1]

**3.1 Connected neural network**

The number of epochs were kept at 100 initially but as can be seen in figure 5, the accuracy peaked way before, therefore the number of epochs were reduced to 50. With little complication we were able to obtain near perfect accuracy and minimal loss for both training and validation. You can check my model here [13].

The final accuracy of the model was determined to be **97.48%**

![Chart

Description automatically generated]()![Chart

Description automatically generated]()

*Figure 5: Model accuracy (left) and Model loss (right) of the connected neural network.*

**3.2 Random forest classifier**

We were able to get a complete classification report for random forest, including the confusion matrix and the accuracy as shown below:

![A picture containing text, receipt

Description automatically generated]()

It can be seen from the report that the model was able to classify all but 4 data points correctly (check confusion matrix), from this the accuracy of the model was determined to be **99.54%.** The model also had a high precision of **0.88** choosing the habitable planets (class 1). What gives random forest superiority over connected neural network for this problem is its ability to provide us a feature importance chart (see Figure 6). Which clearly shows how each of the features played a role in formulating the decisions of our predictive model.

![A picture containing text, newspaper, receipt

Description automatically generated]()![Chart, bar chart

Description automatically generated]()

*Figure 6: The feature importance chart of Random forest (right), feature description and scores (left)*

It can be seen from the chart that random forest gave the more importance to planet’s radius (0.20), and planet’s equilibrium temperature (0.15). Overall planetary features (0 to 5) were given higher importance than the stellar features (6 to 11). You can check my model here. [14]

**3.3 Ada boost**

The classification report for Ada boost is as follows:

![A picture containing text, receipt

Description automatically generated]()

It can be seen from the confusion matrix that ada boost only made 3 wrong predictions and had an overall accuracy of **99.66%.** With a precision of **0.92** when selecting habitable planets from the data. Overall, the boosting method of Adaboost proved to be more accurate and precise in its predictions than the bagging method that was employed by the Random forest. Some interesting results were given by its feature importance chart (see Figure 7).

![A picture containing text, indoor, receipt

Description automatically generated]()![Chart, bar chart

Description automatically generated]()

*Figure 7: The feature importance chart of Ada boost (right), feature description and scores (left)*

Contrary to random forest, ada boost’s predictive model gave more weightage to planet’s flux (solar insolation) and planet’s effective temperature. Moreover, it also gave planet’s period and star’s temperature much higher weightage. But again, on the whole planetary features were dominant in determining habitability. You can check my model here. [15]

**3.4 XG boost**

XG boost gave most promising and astonishing results. Its classification report is presented below:

![A picture containing text, receipt

Description automatically generated]()

From the confusion matrix it can be seen that xg boost only got one prediction wrong in the testing set and correctly classified all others, getting an amazing accuracy of **99.89%.** It also exhibited perfect precision while classifying datapoints into each of the classes. Even more intriguing was its feature importance chart (see Figure 8).

![A picture containing text, newspaper, receipt

Description automatically generated]()![Chart, bar chart

Description automatically generated]()

*Figure 8: The feature importance chart of XG boost (right), feature description and scores (left)*

Interestingly, XG boost gave planet’s radius more weightage than all other features combined (0.51), indicating that it was the single most important feature used by its predictive model during classification. After this, planet’s flux was given most consideration (0.166). This interesting relation between these two features is discussed in the next section. You can check my model here. [16]

# 4 DISCUSSION

# Comparing the results of all the models considered, we can conclude that giving the best results, XG boost is by far the best classification technique for habitability (highest accuracy and precision) that uses the best of both the bagging and boosting methods (see Table 2).

# 

*Table 2: Accuracy comparison of different ML models used*

Given the amazing performance of XG boost model, it’s interesting to note how it assigned feature importance. Planet’s radius and planet’s flux alone counted for around **68%** of the total weightage. So, it’s interesting to see how just based on these two features we can greatly narrow down our search for habitable planets. As is highlighted in the figure we plotted below with solar insolation on the y-axis and planet’s radius on the x-axis (see Figure 9). The graph was plotted using the original exoplanet dataset. And the features are given in comparison to Earth (in Earth units).

*.*Chart, diagram

Description automatically generated

*Figure 9: Solar insulation vs Planet’s radius. Blue = Habitable planets, Orange = Non- habitable planets*

From the graph it can be clearly seen that the exoplanets (orange dots) can have solar insolation all the way up to 4000 times that of the Earth. And their radius can be anywhere from 0 to 20 Earth radii. However, the planets classified as habitable (blue dots) are only found within a very small specific region (as shown in the zoomed part of the figure). We can see that all the blue points lie approximately within **0.2 to 1.6 Earth fluxes**, and within **0.8 to 2.8 Earth radii.** And any point outside this region has been marked as unhabitable. This is most important finding of this project.

# 4.1 Improvements and future outlook

The greatest limitation in this project is the lack of data set available of habitable exoplanets. As only 55 of 4049 confirmed exoplanets were marked as habitable. This results in more than 98% of the data comprising of non-habitable planets and less than 2% marked as habitable. This highly uneven percentage of the two classes makes it a highly unfavorable data set for the machine learning models to learn from. Ideally for classification problems we would want around the same percentage of each classification class in the training dataset.

Secondly, the data is incomplete. We are missing data of a lot of features for exoplanets that play an important role in determining habitability (as shown in figure 3). We need to collect information on all these features to get a complete dataset and the models trained on this data will give more realistic classification of habitability. However, we could see that only using a handful features the models were able to distinguish habitable planets with great accuracy.

Moreover, just a few thousand data points are not enough for the machine learning models to learn properly, usually machine learning models learn from hundreds of thousands of classified datapoints to come up with a strong predictive model. Therefore, we definitely need a bigger dataset, we need to greatly expand our catalog of exoplanets. And launch of satellites like TESS (transiting exoplanet survey satellite) whose sole purpose is to look for exoplanets in foreign star systems, is definitely a giant leap in the right direction. The data we used for this project was obtained from Kepler space telescope which focused on just a small part of the sky, however TESS will survey the whole northern sky for exoplanets, so we can expect to find a large addition in the exoplanet catalog in the future. However, that won’t be enough we need to eliminate false positives and also classify the remaining data according to habitability so that machine learning models can learn from them. Once we’ll have a big enough classified dataset, then we can automate the task of habitability classification using machine learning. And as it was evident in this project, machine learning models are very effective for this purpose.

# 5 REFERENCES

[1] Kremer, J., Stensbo-Smidt, K., Gieseke, F., Pedersen, K. S. & Igel, C. Big Universe, Big Data: Machine Learning and Image Analysis for Astronomy. *IEEE Intell. Syst.* **32**, 16–22 (2017).

[2] NASA exoplanet archive: https://exoplanetarchive.ipac.caltech.edu/docs/data.html

[3] Shallue, C. J. & Vanderburg, A. Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. *AJ* **155**, 94 (2018).

[4] PHL’s catalog: http://phl.upr.edu/projects/habitable-exoplanets-catalog/data/database

[5] Basak, S., et al. Habitability Classification of Exoplanets: A Machine Learning Insight. arXiv:1805.08810 **[astro-ph.IM]**(2018)

[6] Hora, K. Classifying Exoplanets as Potentially Habitable Using Machine Learning. in *Advances in Intelligent Systems and Computing*, 203–212 (Springer, Singapore, 2017).

[7] Deeg, H. J. *Handbook of exoplanets* (Springer, 2018)

[8] Data I prepared: https://drive.google.com/file/d/1NGqsiWpgttZYx7q5pPH2gCdB5fH4DSvq/view?usp=sharing

[9] Huang, G., et al. Densely Connected Convolutional Networks. arXiv:1608.06993 **[cs.CV]**(2017)

[10] Breiman, L. Random Forests. *Machine Learning* **45,**5–32 (2001).

[11] Freund, Y., & Schapire, R. E. A Short Introduction to Boosting. in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1401–1406 (2019).

[12] Chen, T. & Guestrin, C. XGBoost. in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2016). doi:10.1145/2939672.2939785

[13] Connected neural network: https://colab.research.google.com/drive/1Rafzt4KtIL95mSmSnBlyN0cooxlaYFhb?usp=sharing

[14] Random forest: https://colab.research.google.com/drive/1G2LI9CKLBYXKsuZp9rgi3vVNBDeUITlU?usp=sharing

[15] Ada boost: https://colab.research.google.com/drive/1O8LyDxvb9BCRREVmIw4h9WfQkrL4bKL2?usp=sharing

[16] XG boost: https://colab.research.google.com/drive/1WdNpidqKl7HYXQbJ9KNpoT9pPiD3GiFh?usp=sharing