SÃO PAULO STATE UNIVERSITY (UNESP)
CAMPUS DE GUARATINGUETÁ

Tiago Francisco Lins Leal Pinheiro

Supervised learning for predicting stable orbital maps in the Three-body problem

Guaratinguetá/Brazil

2024

Tiago Francisco Lins Leal Pinheiro

Supervised learning for predicting stable orbital maps in the Three-body problem

Doctoral Thesis presented to the Graduate Program in Physics and Astronomy of the Sao Paolo State University "Júlio de Mesquita Filho" as a partial requirement to obtain the degree of Doctor in Physics and Astronomy

São Paulo State University (UNESP)
Graduate Program in Physics and Astronomy

Supervisor: Prof Dr Rafael Sfair

Guaratinguetá/Brazil
2024

P654s

Pinheiro, Tiago Francisco Lins Leal
Supervised learning for predicting stable orbital maps in the Three-body problem / Tiago Francisco Lins Leal Pinheiro - Guaratinguetá, 2024.
118 f : il.
Bibliografia: f. 90-96

Tese (Doutorado) – Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2025.

Orientador: Prof. Dr. Rafael Sfair

1. Exoplanetas. 2. Planetas. 3. Aprendizado do computador. I. Título.

CDU 523.4(043)

Luciana Máximo
Bibliotecária/CRB-8 3595

unesp **_UNIVERSIDADE ESTADUAL PAULISTA_**
CAMPUS DE GUARATINGUETÁ

TIAGO FRANCISCO LINS LEAL PINHEIRO

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE
"DOUTOR EM FÍSICA"

PROGRAMA: FÍSICA E ASTRONOMIA
CURSO: DOUTORADO

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

gov.br Documento assinado digitalmente
ERNESTO VIEIRA NETO
Data: 18/11/2024 16:25:37-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Ernesto Vieira Neto
Coordenador

_B A N C A   E X A M I N A D O R A:_

gov.br Documento assinado digitalmente
RAFAEL SFAIR DE OLIVEIRA
Data: 11/11/2024 16:01:24-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. RAFAEL SFAIR DE OLIVEIRA
Orientador - UNESP

gov.br Documento assinado digitalmente
VALERIO CARRUBA
Data: 11/11/2024 12:03:36-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. VALERIO CARRUBA
UNESP

gov.br Documento assinado digitalmente
ROSANA APARECIDA NOGUEIRA DE ARAUJO
Data: 11/11/2024 14:58:55-0300
Verifique em https://validar.iti.gov.br

Profa. Dra. ROSANA APARECIDA NOGUEIRA DE ARAUJO
UNESP

gov.br Documento assinado digitalmente
ELBERT EINSTEIN NEHRER MACAU
Data: 06/11/2024 16:03:15-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. ELBERT EINSTEIN NEHRER MACAU
UNIFESP

gov.br Documento assinado digitalmente
BRUNO EDUARDO MORGADO
Data: 11/11/2024 06:01:29-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. BRUNO EDUARDO MORGADO
UFRJ

CURRICULAR DATA

TIAGO FRANCISCO LINS LEAL PINHEIRO

BIRTH	03/06/1993 - Lorena/SP
FILIATION	Roberto Carlos Pinheiro Jurema Lins Leal Pinheiro
2012 - 2016	Bachelor in Physics University of São Paulo State University (UNESP)
2017 - 2019	Master in Physics University of São Paulo State University (UNESP)
2019 - 2024	Doctorate in Physics University of São Paulo State University (UNESP)

Acknowledgements

First, I would like to thank my parents and my family, especially my mom Jurema and my father Roberto, my siblings Maria Teresa (Tête), Daniel José (Nel), Gabriel Augusto (Tu), and Rafael Alberto (Grudinho), my brother-in-law Clayton (Jamal), and my niece Giovana Maria (Gi) for their constant support and encouragement in all my decisions.

Next, I would like to thank my supervisor, Dr. Rafael Sfair, for teaching and supervising me since my undergraduate studies. His advice has made me the professional I am today.

I would like to thank FEG's professors for their teaching, collaboration on scientific projects, and friendship: Dr. André Amarante, Dr. Ernesto Vieira, Dr. Elias, Dr. Othon Winter, Dra. Olivia, Dr. Rafael Ribeiro, and Dra. Silvia Winter.

My special and heartfelt thanks to all my friends who have supported me throughout my academic journey, who laughed with me and made me laugh: Ana Carla, Ana Luiza (Aninha), Anderson, Andreza "Dara"(Flecha), Bárbara Banheti (Babi), Bruno (Cachu), Diego (Raj), Douglas (Dompo), Giovana Ramon (Cosmo), who also collaborate with me in the projects of this text, Gabriel (Bahia), José Eduardo (Dudu), Juan, Juliana Paiva (Juh), João Miguel, João Victor, Lucas Brito (Eminem), Lucas Ferreira, Luiz Gomes (Capão), Luiz Gustavo, Kelvin, Marcos, Maria Eduarda (Duda), Mateus (Canário), Milena, Millena, Natália (Naty), Nilce, Omar, Patricia Buzzatto (Paty), Pedro, Rai, Raíssa (Toy), Rodrigo (Lineu), Rosimara (Rosi) and Vanessa (Vans). My most sincere thanks to my friends Luana Liberato, Gustavo Madeira, and Victor Lattari for being always with me since the first year of undergraduation. Thank you for all the partnership, the jokes, the laughter, the gossip, and the hangouts.

I sincerely thank the selected committee and substitute members for taking time to evaluate my thesis.

Paraphrasing Snoop Dogg's speech when receiving his star on the Hollywood Walk of Fame:

"Last but not least, I wanna thank me. I wanna thank me for believing in me. I wanna thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting. I wanna thank me for always being a giver and trying to give more than I receive. I wanna thank me for trying to do more right than wrong. I wanna thank me for just being me at all times. Tiago, you are amazing."

This study was financially supported by:

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001

Fundação de Amparo a Pesquisa no Estado de São Paulo (FAPESP) - Proc. 2016/24561-0

German Research Foundation (DFG) - Project 446102036

"The measure of love is to love without measure"

Saint Augustine

Abstract

Numerical simulations of N-body systems are often used to map stable regions around planets, stars or even small bodies. Which can provide insights into the potential presence of planets in the case of star, satellites and ring systems.

In this text we present an alternative approach to speed up the numerical results. We used an application of machine learning techniques to predict the orbital stability of test particles around the primary and secondary body in three bodies problem.

In the first work, we aim to predict stability maps around planet. we generate a dataset with 100,000 different initial conditions of the elliptic three-body problem involving a star, planet, and a test particle, using a wide range of orbital and physical parameters. Our machine learning models, particularly with XGBoost algorithms, achieved an accuracy of 98.48%. We also validate our model reproducing prior numerical studies. This demonstrates that machine learning can efficiently reduce the computational time efforts.

In the second work, we applied machine learning to analyze the planetary ring systems around J1407b and PDS110b. Our models constraints the range of planetary mass and eccentricity, as well the ring inclination and radius. For PDS110b, a star-planet system with $\mu > 0.04$ and planet with eccentricity < 0.3, the ring inclination is $\geq 150°$, and the ring radius is between 0.2 and 0.3, matching the observed eclipses with probability $< 7.2\%$. For J1407b, a planet with a semi-major axis of 3–15 au and eccentricity exceeding 0.7, therefore any of these case could explain the observable transverse velocity.

In the last work, we explore the stability of the sailboat region in binary systems. Our machine learning models achieved an accuracy of over 97% in classifying particle, we also use Poincaré Surface of Section to confirm the stability of this region. Our results suggest that the sailboat region exists for binary mass ratios $\mu = [0.05 - 0.22]$. This region is very sensitive to variations in the eccentricity of the secondary body; however, it remains robust for large particle inclinations. The sailboat region exist within two intervals of the particle's argument of pericenter.

In all cases, the machine learning models significantly reduced computational time, enabling the exploration of vast parameter spaces while achieving highly accurate performance.

Keywords: machine learning - stability maps - exoplanets - exorings - binary system

Resumo

Simulações numéricas de sistemas N-corpos são frequentemente usadas para mapear regiões estáveis ao redor de planetas, estrelas ou até pequenos corpos. Através dessas simualações é possível detectar regiões estável que poderiam conter planetas no caso de estrelas, satélites e até sistemas de anéis.

Neste texto será apresentado uma abordagem alternativa para acelerar os resultados das simulações numerical. Utilizando uma aplicação com machine learning é possível estimar a estabilidade orbital de partículas de teste ao redor dos corpos primário e secundário no problema do problema de três corpos.

O primeiro trabalho apresentado tem como objetivo utilizar machine learning para prever mapas de estabilidade ao redor de planetas. Foram realizadas 100.000 simulações numéricas do problema dos três corpos elíptico, contendo uma estrela, um planeta e uma partícula de teste, utilizando uma ampla gama de parâmetros orbitais e físicos. Os modelos de machine learning apresentaram ótimas performances, e, em particular, o algoritmo XGBoost alcançou uma acurácia de 98,48%. O modelo campeão foi utilizado para reproduzir os resultados de artigos que utilizaram simulações numéricas. Isso demonstrou que o machine learning pode reduzir, de maneira eficiente, o tempo computacional das simulações numéricas.

No segundo trabalho, aplicamos machine learning para analisar os sistemas de anéis planetários ao redor de J1407b e PDS110b. O objetivo desse trabalho era restringir parametros de massa e excentricidade do planeta, inclinação e raio do anel. No caso PDS110b, um sistema estrela-planeta com $\mu > 0{,}04$ e excentricidade planetária $< 0{,}3$, um anel com inclinação $\geq 150°$ e um tamanho radial entre 0,2 até 0,3 au. consegue reproduzir o eclipse observado, com uma probabilidade $<$ 7,2%. No caso do sistema J1407b, um planeta com semi-eixo maior de 3–15 au e excentricidade superior a 0,7 pode explicar o eclipse observado, entretanto, não foi possível reproduzir um transito one a velociada do planeta possui a mesma velocidade transversal observada.

O último trabalho apresentado, tem como objetivo analisar a estabilidade da região "sailboat" em sistemas binários. Modelos de machine learning alcançaram uma acurácia superior a 97% na classificação de partículas, e também utilizamos a Superfície de Secção de Poincaré para confirmar a estabilidade dessa região. Os resultados sugerem que a região "sailboat" existe para razões de massa binária μ = [0,05 - 0,22]. Essa região é muito sensível a variações na excentricidade do corpo secundário; no entanto, ela é robusta para grandes variações na inclinações das partículas. A região "sailboat" existe dentro de dois intervalos do argumento do pericentro das partículas.

Em todos os casos, os modelos de machine learning reduziram significativamente o tempo computacional, possibilitando a exploração de uma vasta gama de parâmetros iniciais, além de alcançarem um ótimo desempenho em suas performances.

Palavras-chave: machine learning - mapas de estabilidade - exoplanetas - exoanéis - sistemas binários

List of Figures

List of Tables

Contents

1 Introduction

The proposal of this work is to apply machine learning techniques to classify stable particles in the three-body problem. Through this approach, we aim to predict stability maps around hypothetical planets and identify stable regions within binary systems, such as the sailboat region. By leveraging machine learning, we can significantly speed up the numerical results, enabling to explore unfeasible quantity of initial conditions.

This thesis is divided into three distinct projects, each presented in a separate chapter. The chapters are structured in such a way that they can be read independently, allowing the reader to focus on each project without needing to follow a strict sequence.

This work includes the collaboration of Giovana Ramon, who focused on optimizing the machine learning model's performance; Ernesto Vieira, who contributed to the Poincaré surface of section; and Rafael Sfair, who worked on the public web interface and also contributed to the numerical results and the machine learning model.

The structure of the text is organized as follow:

Chapter 2 provides a comprehensive review of key machine learning concepts, with a specific focus on classification tasks, which are the approach used in this thesis. In this chapter, we explain how several tree-based algorithms work, address methods for handling imbalanced datasets, and explore approaches to configuring machine learning models for optimal performance. We also describe how to calculate the metrics used to evaluate these models.

Chapter 3 presents a machine learning approach to predicting stability regions around planet using data generated from numerical simulations of the elliptical three-body problem. The machine learning model can then efficiently predict orbital stability outcomes for test particle orbiting a planet and gravitational disturb by star. The chapter discusses the machine learning techniques applied, evaluates model performance, and compares results with traditional studies, showing that this approach offers a significant reduction in computational time while maintaining high predictive accuracy.

Chapter 4 investigates the potential existence and characteristics of exoplanetary ring systems through the analysis of dimming events observed in PDS110 and J1407 stars. By employing a machine learning model to predict the stability of particles in these systems, the study explores several initial conditions to constrain the mass, eccentricity of host planet, as well the size, and inclination of the rings. The findings suggest plausible ring sizes and properties for exoring candidates and estimate the likelihood of reproducing the observed dimming events, offering insights into the nature and detectability of exoplanetary ring systems.

In the Chapter 5 studies the stability of the sailboat region, a peculiar stable region present in binary systems where the the particle are located at middle separation distance between the missives bodies and have eccentric orbits. We explore how changes in parameters such the binary mass ratio, secondary body eccentricity, particle orbital inclination, and argument of pericenter affect the existence and size of the sailboat region. By conducting extensive numerical simulations and employing a machine learning model, The findings reveal the sailboat's resilience into the conditions under which the sailboat region remains stable.

The last chapter contains the final remarks of all three projects. The appendices include copies of both published and submitted papers, which can serve as a quick reference source.

2 Fundamentals of Classification Learning

Machine Learning is the science that enables a computer to execute specific tasks without being explicitly programmed for it (Samuel, 1959). There are numerous situations to apply machine learning techniques, such as problems require significant manual adjustments or long lists of rules, to find possible solutions for complex problems where traditional approaches not work well, and to provide insights from complex issues and large datasets (Géron, 2022).

Figure 1 illustrate a flowchart about Machine Learning (ML) approach to solve a problem. First step is **study the problem**, which consist in identifying what needs to be solved or analyzed. Second step is **train ML algorithm** through the dataset, once the algorithm is trained, it produces a **solution**. The third step is **inspect the solution**, which consist in evaluating the performance of machine learning model. Last step is **understand the problem better**, the user gains a better understanding of the problem. If the solution is not satisfactory, the process can be repeated.

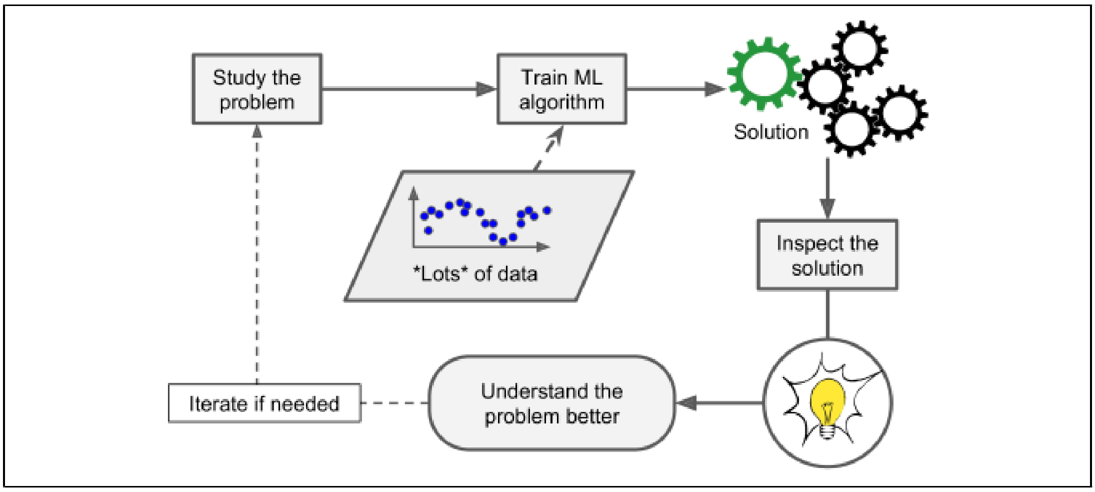
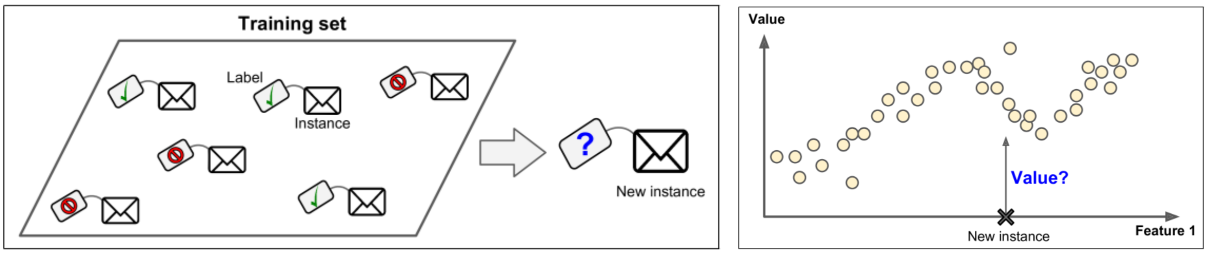


Figure 1 – A flowchart illustrating the Machine Learning approach (Géron, 2022).

The main areas in the science of machine learning include supervised learning, unsupervised learning, and reinforcement learning. These approaches may or not use a dataset during the learning process (Géron, 2022).

Supervised learning is characterized by having a supervisor during the training stage. This technique uses an input dataset consisting of features and corresponding labels for a specific task. The machine learning algorithm utilizes this data to generalize the relation between the features and labels, enabling it to predict outcomes for unknown data (Shalev-Shwartz and Ben-David, 2014b). When these labels consist of continuous numerical values, it is a regression task; if the labels are discrete values representing different classes, it is a classification task.

The figure 2 illustrates examples of supervised learning for different tasks. The left panel shows classification, where the algorithm determines whether emails are spam or not. In contrast, the right panel shows regression, where the algorithm fits the dataset and predicts values for unknown data.



Classification task Regression task

Figure 2 – Representation of classification and regression task (Géron, 2022).

Unsupervised learning does not include labels in the dataset. Instead, the algorithm analyzes the data with the aim of producing a generalization for labels, so the algorithm groups data into clusters of similar features (Shalev-Shwartz and Ben-David, 2014b). Figure 3 shows an example of clustering: on the left, the dataset is represented, while on the right, the data is divided based on the similarity of samples (or instances) features.

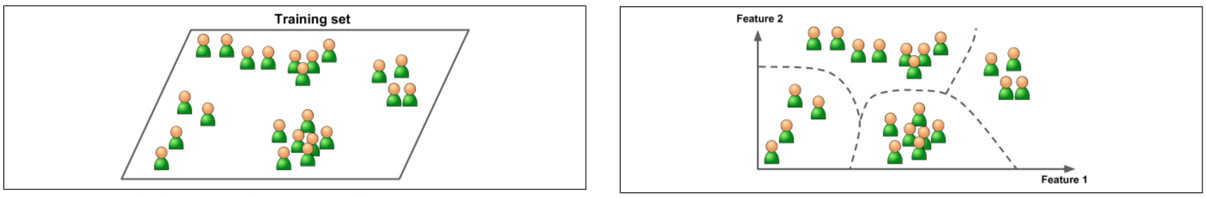


Figure 3 – Representation of clustering task (Géron, 2022).

Reinforcement learning is a learning approach wherein an agent learns behavior by interacting with a dynamic environment through trial-and-error (Kaelbling et al., 1996). Reinforcement learning involves figuring out how to choose actions based on situations to achieve the highest possible reward. Instead of being directed on which actions to take, the learner explores various actions (through trial-and-error) to see which ones generate the most reward. In more complex scenarios, actions can influence not just the immediate reward but also the future states and the rewards that follow (Sutton, 2018).

Figure 4 shows an example of trial-and-error process. The agent, represented by the robot, interacts with the environment, which presents two possible actions. In the case shown, the agent is penalized for its actions, which makes the model never take this repeat this action again.

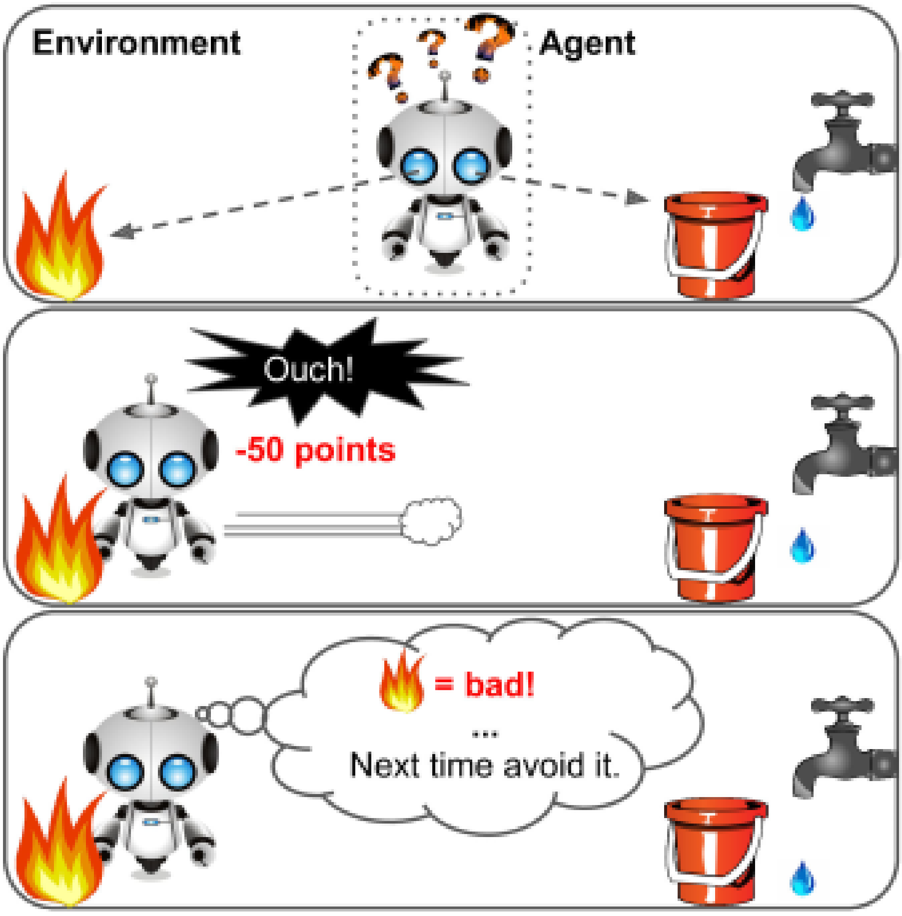


Figure 4 – An example of a trial-and-error process in reinforcement learning approaches (Géron, 2022).

Several studies have highlighted the effectiveness of machine learning algorithms in processing astronomical data. Carruba et al. (2021) applied Artificial Neural Networks (ANNs) to classify asteroids in the M1:2 mean-motion resonance with Mars, demonstrating the potential of ML for resonance studies. In a more recent work, the same group, in collaboration with the Dark Energy Survey (DES), extracted taxonomical information for main-belt asteroids using ML techniques (Carruba et al., 2024). Additionally, Carruba et al. (2022) provided an extensive review of the application of machine learning in asteroid dynamics, discussing how these methods can enhance the understanding of asteroid orbits and interactions.

This chapter is organized as follows: section 2.1 introduces the main concepts of classification algorithms and the machine learning tasks applied in our work. Section 2.2 addresses the with imbalanced datasets. Sections 2.3 and 2.4 present techniques for optimizing algorithm configurations, while section 2.5 details the math used to evaluate the performance of machine learning models in classification tasks.

2.1 Classification Algorithms

Classification tasks involve using training data during the learning process to create a model that generalizes classes and predicts outcomes for new unseen instances. In this text, we address binary classifications, where the objective is to categorize data points into one of two distinct classes, often labeled as 0 and 1, or negative and positive. in our case stable and unstable particles.

The training data is utilized by the algorithm to identify the optimal generalization features for different classes. The validation data provides an unbiased assessment of the model's performance, aiding in the tuning of hyperparameters and thresholds, selecting the most effective machine learning algorithm, and identifying underfitting, where the model fails to learn the data's structure, and overfitting, when learns the details of the training data too well, leading to poor generalization on new data (Shalev-Shwartz and Ben-David, 2014a).

The cross-validation technique involves partitioning the training data into subsets. Subsequently, we train and evaluate the model on distinct subsets, followed by averaging the outcomes to estimate the true error of the model (Fushiki, 2011).

After training several models with different configurations multiple times on the reduced training set and evaluating them through cross-validation, we identified the best-performing model. Subsequently, we retrained the selected model using the complete training dataset and evaluated using unseen data, the test dataset.

The figure 5 illustrates the process of identifying the best-performing machine learning model using cross-validation, where the dataset is divided into three main parts: training, validation, and testing. The training data (blue) is further split into multiple folds, with one fold serving as the validation set (red) in each iteration while the remaining folds are used for training (green). This rotation ensures that every fold is used for validation exactly once, and the performance is evaluated through validation scores. After completing the cross-validation, the model is tested on the independent test set (yellow) to calculate the test score, providing an unbiased assessment of its generalization capability.

The projects presented in this text employed eight distinct machine learning algorithms to find the optimal approach: Nearest Neighbors, Naive Bayes, Artificial Neural Network, and tree-based methods Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost), Histogram Gradient Boosting, and Light Gradient Boosting Machine (LightGBM). Lately, we ran a Genetic algorithm to verify our findings.

Given that all tree-based algorithms demonstrated better and comparable performance, we will focus our analysis on these methods. Except for the Decision Tree, all other classifiers are ensemble methods, which utilize a group of weaker learners (in our case, an ensemble of decision trees) to improve their predictions. Collectively, a group of predictors

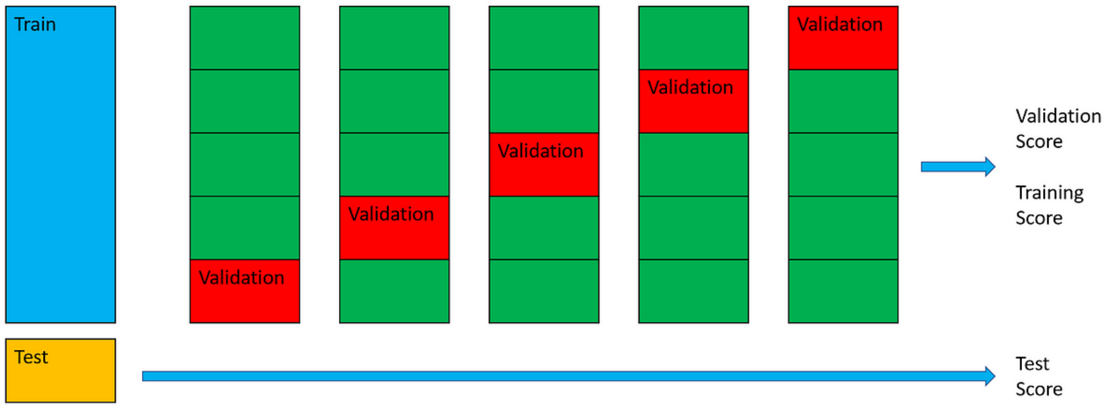


Figure 5 – Cross validation approach.

tends to make more accurate predictions than a single predictor (Géron, 2022). In the following sections, each of these algorithms will be described in detail.

2.1.1 Decision Tree (DT)

Decision trees, first introduced by Quinlan (1986), are algorithms used to classify instances by progressively splitting the data into subsets at decision points called nodes. A node represents a point where the data is divided based on a specific condition or rule, such as $x_i < \mathrm{t}_i$, where x_i is a feature, and t_i is the threshold value applied to that feature (Shalev-Shwartz and Ben-David, 2014b). The process continues until the data cannot be split further or a stopping criterion is met. The final subdivisions of the tree are called leaf nodes, which represent the outcomes or classes assigned to the instances within that subset. Essentially, the leaf nodes are terminal points of the tree, containing predictions or classifications derived from the features and rules applied in the preceding nodes (Learning, 1997).

The tree begins with the root node, where the data is split into two subsets, creating two new nodes that can either be leaf nodes or decision nodes. A leaf, or terminal node, is where an instance is classified, while a decision node further splits the data into two new subsets and nodes. This process repeats until the branch reaches its terminal nodes.

In the figure 6 shows a sketch of decision tree method and its nomenclature.

The optimal threshold value for each node in a decision tree is determined by evaluating the impurity of the resulting subsets after splitting. The goal is to select a feature and threshold combination that minimizes a specific cost function for classification, denoted as $(J(k, t_i))$. This cost function quantifies impurity of the two subsets created by the split and it is expressed mathematically as:

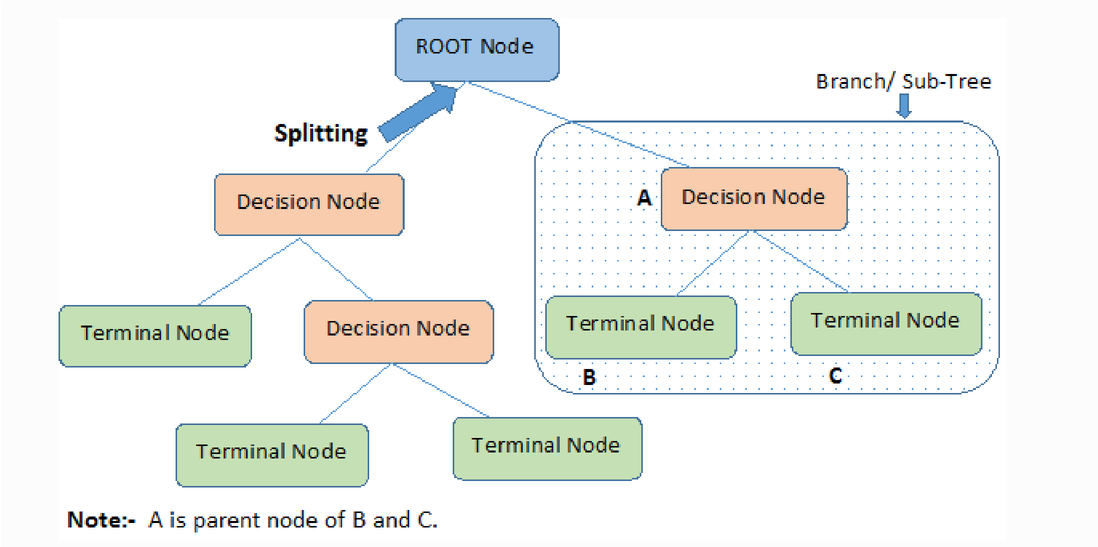


Figure 6 – Decision tree representation.

$$J(k, t_i) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}, E - CostFunction \tag{2.1}$$

where m represents the total number of instances in the node, m_{left} and m_{right} denote the quantities of instances in the left and right subsets, while G_{left} and G_{right} are the Gini impurity scores for these subsets, respectively.

The Gini impurity of a node (G_i) measures the probability of misclassifying a randomly chosen instance from the subset if it were labeled according to the distribution of classes in that subset. It is defined as:

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2 E - Gini \tag{2.2}$$

where n is the total number of classes, and is the probability of an instance in subset i belonging to class k.

The probability $p_{i,k}$ is calculated as the ratio of the number of instances in subset i that belong to class k $(m_{i,k})$ to the total number of instances in the subset (m_i):

$$p_{i,k} = \frac{m_{i,k}}{m_i} .E - Probability \tag{2.3}$$

An alternative method for estimating node impurity is to calculate the entropy. Entropy provides a measure of uncertainty or disorder in the class distribution of a node and is defined as:

$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^{n} p_{i,k} \log_2 (p_{i,k}). \tag{2.4}$$

Both Gini impurity and entropy are commonly used criteria for building decision trees, as they aim to reduce the impurity of the nodes at each split. In practice, the two criteria often yield similar tree structures (Géron, 2022). However, Gini impurity is computationally less intensive because it does not involve logarithmic calculations, making it faster to compute.

2.1.2 Random Forest (RF)

Occasionally, it can be useful to create a set of weaker classifiers to improve the overall performance of the model. When this approach involves training multiple instances of the same machine learning algorithm on different random subsets of the dataset, it is known as bagging, or Bootstrap Aggregating (Géron, 2022).

A Bagging Classifier creates random subsets of the dataset and trains a model on each subset independently (Breiman, 1996). The final result is obtained by combining the predictions of all the models using a voting scheme, thereby generally improving the robustness and accuracy of the overall prediction. This process exemplifies ensemble learning approach, which use the concept of wisdom of the crowd suggests that the collective judgment of diverse, independent individuals often outperforms that of a single expert (Sagi and Rokach, 2018).

An example of a bagging classifier is the Random Forest, introduced by Breiman (2001), which is an ensemble of Decision Trees. The use of Random Forest reduces the risk of overfitting, as the method helps decrease the correlation between the trees, what would lead to more robust predictions (Shalev-Shwartz and Ben-David, 2014b). Figure 7 illustrates the training process of a Random Forest.

This algorithm trains numerous decision trees using different subsets of the training data through a bootstrap process, which involves random sampling of the dataset with replacement. By sampling with replacement, some instances may be repeated in each subset. The final prediction is determined either by the majority vote of the outputs from these individual decision trees or by averaging their probability distributions for a more nuanced classification. The figure 8 shows an example of how bootstrap method works. In this example, there are 20 different names in the original dataset, although when randomly selecting a group of names (referred as the bootstrap dataset), some names are repeated twice (blue) and other three times (red).

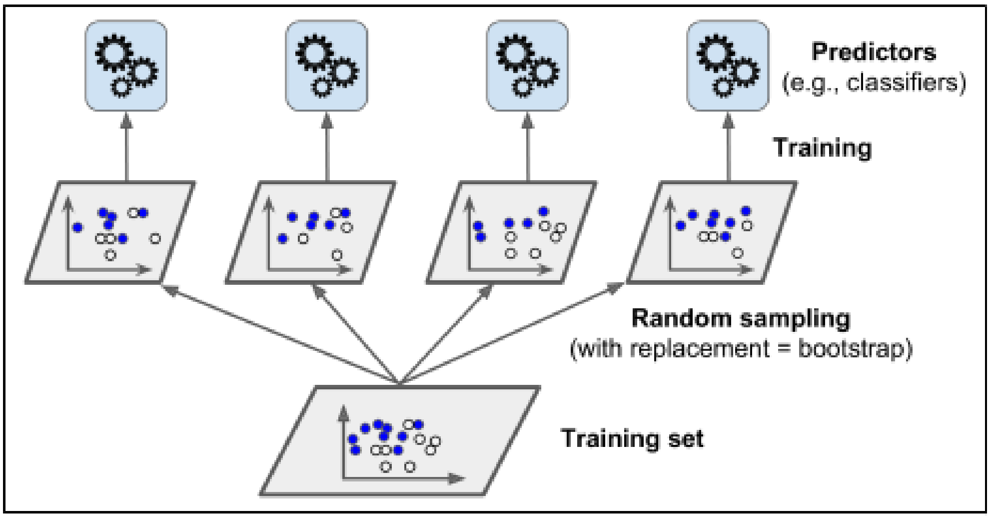


Figure 7 – Random Forest algorithm representation (Géron, 2022).

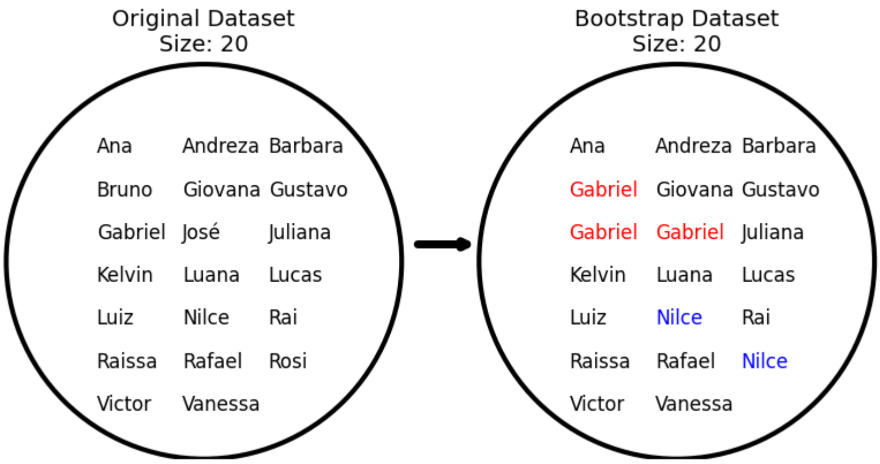


Figure 8 – Simple example showing how the bootstrap method works.

2.1.3 Boosting algorithms

Boosting algorithms combine several weak learners to create a strong learner by training predictors sequentially. Each predictor rectify the errors made by its predecessor (Géron, 2022). This is achieved by using the residuals from the previous model to fit the next model. In the following section, we will provide an overview of three boosting algorithms: XGBoost, Histogram Gradient Boosting, and LightGBM.

2.1.3.1 Extreme Gradient Boosting (XGBoost)

XGBoost (Extreme Gradient Boosting) is a ensemble of decision trees learners, introduced by Chen and Guestrin (2016), based on the idea of Gradient-Boosted Tree (GBT) by Friedman (2001). This method involves computing the residuals, the differences between the actual outcomes and the model's predictions to guide the construction of subsequent trees by focusing on the errors made in previous iterations, thereby iteratively improving the model's performance (Wade and Glynn, 2020). Unlike the traditional sequential approach of Gradient-Boosted Trees, XGBoost can constructs multiple trees in parallel by partitioning the data into subsets during each iteration.

In contrast to Random Forest, which trains trees independently, XGBoost sequentially incorporates new trees to address the residuals left by preceding trees. The final prediction is then derived from the combination of these trees' outputs, each weighted differently (Chen and Guestrin, 2016).

2.1.3.2 Light Gradient-Boosting Machine (LightGBM)

Light Gradient Boosting Machine (LightGBM) is a variant of the Gradient-Boosted Tree algorithm, introduced in 2016 as part of Microsoft's Distributed Machine Learning Toolkit (DMTK) project (Ke et al., 2017). This method uses histograms to discretize continuous features by grouping them into distinct bins. LightGBM use leaf-wise growth strategy, which allows the tree grows vertically as shown in figure 9. This approach also results in a faster convergence and lower residuals.

Figure 9 illustrates the difference between the leaf-wise growth strategy used by LightGBM and the level-wise growth strategy used by XGBoost. The orange nodes in the diagram indicate where splits occur during tree growth at each step.

In the level-wise growth, the tree expands level by level, splitting all nodes in the same level before moving to the next. This method, while balancing the tree, may not reduce loss optimally since it does not always consider the best global split point. As a result, depth-wise growth tends to overfitting less, as it produces shallower trees.

On the other hand, LightGBM's leaf-wise growth expands the tree by splitting the node with the highest split gain, irrespective of its depth. This approach can reduce loss more effectively than depth-wise growth because it always selects the most optimal split point. However, it might lead to overfitting as it produces more complex and deeper trees.

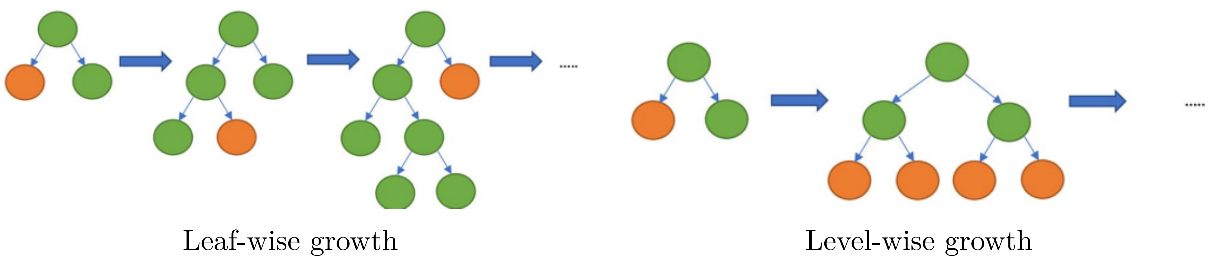


Figure 9 – Representation of leaf and level wise growth (Narimani et al., 2023).

] the comparison between both method may be summarized as:

1. **Split Priority:**
    - Level-wise growth considers all nodes at the same level, promoting uniform tree structure.
    - Leaf-wise growth focuses solely on the node with the highest split gain, optimizing loss reduction but sacrificing balance.
2. **Tree Depth and Complexity:**
    - Level-wise growth generates shallower and more interpretable trees.
    - Leaf-wise growth often results in deeper trees, capturing more complex data patterns.
3. **Risk of Overfitting:**
    - Level-wise growth is less prone to overfitting due to its regular and shallow structure.
    - Leaf-wise growth can overfit easily without proper regularization, especially on small datasets.
4. **Performance:**
    - Level-wise growth offers robust performance across a variety of datasets.
    - Leaf-wise growth performs exceptionally well in scenarios that require precise and detailed loss reduction.

LightGBM offers different characteristics compared to XGBoost, providing alternative benefits that may be more suitable depending on the specific use case, such as more efficient memory usage, lower computational cost for calculating split gains, and reduced communication overhead during parallel processing, which leads to faster training times. Additionally, the algorithm introduces two innovative techniques: Gradient-Based One-Sided Sampling (GOSS) and Exclusive Feature Bundling (EFB) (Ke et al., 2017). GOSS selects samples to form the training datasets for constructing the base trees, while EFB combines sparse features into a single feature to optimize performance (Bentéjac et al., 2021).

2.1.3.3 Histogram Gradient Boosting (HGB)

Histogram Gradient Boosting (HGB), an algorithm inspired by LightGBM, is designed to be highly efficient for large datasets, often with tens of thousands of instances or more. Unlike traditional decision trees, which can be computationally expensive and slow due to the need to evaluate many possible splits across continuous values (Ibrahim et al., 2023), HGB employs histogram-based techniques to significantly streamline this process.

As shown in Figure 10, HGB discretizes continuous values into a fixed number of bins. This reduction in the number of potential split points that the algorithm needs to assess allows for the use of more efficient integer-based data structures, resulting in faster and more efficient tree-building (Wade and Glynn, 2020).

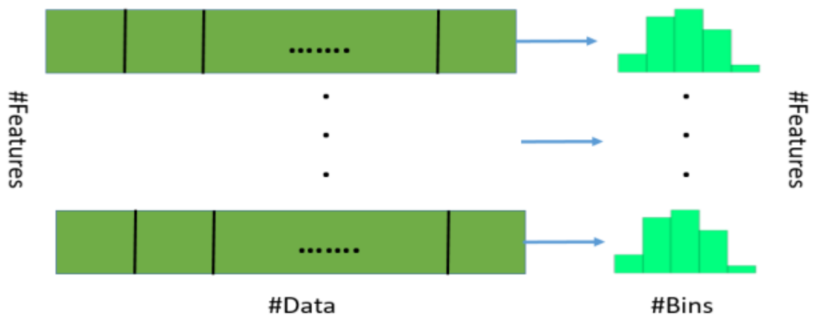


Figure 10 – Converting data into bins (Ibrahim et al., 2023).

To identify the best machine learning model, it is important to follow these steps: analyze the dataset, as an imbalanced class distribution can lead to poor generalization; apply resampling techniques to tackle this issue; evaluate the performance of various machine learning algorithms; and optimize each algorithm by tuning hyperparameters and thresholds. These techniques will be discussed in detail in the following sections.

2.2 Imbalanced Class

An imbalanced dataset occurs when the instances are not evenly distributed among the classes. This imbalance can introduce biases in classification and negatively impact the performance of various machine learning algorithms during the generalization process (Zheng et al., 2015). Many machine learning algorithms assume that classes are evenly distributed, which can lead to poor classification performance, particularly for the minority class (Kumar and Sheshadri, 2012; Carruba et al., 2023).

Resampling the training data is one strategy that can be used to address an imbalanced dataset. Carruba et al. (2023) studied asteroid resonant dynamics and tested the effect of various resampling methods on the performance of machine learning algorithms. Their results showed that a severe case of imbalanced data occurs when the ratio between the minority and majority classes is 1:100. They also concluded that it is always important to verify the effectiveness of resample methods in improving machine learning model performance.

Many datasets presented in this text are imbalanced, with some having a ratio of 1:100. We tested the performance of our machine learning models both with and without the resampling technique.

Undersampling is a technique that removes instances from the majority class. The weakness of this method is potential loss of information by removing part of the data (Mohammed et al., 2020). Another resampling technique is oversampling, which involves increasing the size of the minority class until the classes are balanced.

Here, we evaluated the performance of machine learning models with and without resampling techniques: random undersampling, random oversampling, SMOTE, Borderline SMOTE, and ADASYN.

Random oversampling replicates instances from the minority class until the classes are balanced. However, this approach may lead to overfitting, and alternative solution to avoid it is use Synthetic Minority Over-sampling Technique (SMOTE).

Chawla et al. (2002) proposed oversampling the minority class by generating synthetic instances. First, the method calculates the difference between an instance and its nearest neighbors (instances) in the feature space and this difference is multiplied by a random number (r) between 0 and 1. Finally, a new instance is created by adding this result to the feature vector of the original instance.

Assuming two instances (x_1,y_1) and (x_2,y_2) from the minority class, x and y are features from the dataset, a new synthetic instance (x_3,y_3) will be generate as:

$$x_3 = x_1 + r(x_2 - x_1) \tag{2.5}$$

$$y_3 = y_1 + r(y_2 - y_1) \tag{2.6}$$

Another sampling technique based on SMOTE is Borderline-SMOTE (Han et al., 2005). This method attempt to identify the borderline of each class and avoid to use the outliers (noise points) of minority class to resample them. The algorithm works with the following steps:

1. For each instance of a minority class, check which class belongs their nearest neighbors. Neighbors are other data points in the dataset that are closest to the instance being analyzed. This proximity is determined in the feature space using a Euclidean distance metric.

2. Count the number of closest instances that belong to the majority, if all neighbours are from other classes, this point is classified as noise point, and it is ignored to resample the data.

3. If more than a half of neighbours are from other classes, this is a border point. In this case, the algorithm calculate K nearest neighbours are the same class to generate a synthetic instance between them.

4. If more than a half of neighbours are from the minority, it is a safe point and SMOTE technique is applied normally.

One last oversampling approach presented here is Adaptive Synthetic (ADASYN), which generates synthetic instances for the minority class while considering the weighted distribution of this particular class.

The methodology employed by He et al. (2008) involves an estimation for each minority instance the impurity of its nearest neighbours (I_i), the ratio among K neighbours that are from majority classes is give by:

$$\mathrm{I}_i = \frac{\Delta_i}{\mathrm{K}} \tag{2.7}$$

here, Δ_i is number of non minority class between K nearest neighbours. The subsequent step is normalized the I_i value (equation 2.8) and then multiplying this result by the total number of synthetic instances to be generated. This procedure determines the number of synthetic instances needed to be created for each minority instance.

$$\hat{I}_{\mathrm{i}} = \frac{I_{\mathrm{i}}}{\sum\limits_{i=1}^{\mathrm{m}} I_{\mathrm{i}}} \tag{2.8}$$

where m is size of minority class data.

2.3 Hyperparameters Tuning

Hyperparameters are specific algorithm parameters that must be set before the training process, and the model's performance directly depends on them (Weerts et al., 2020). The process of tuning these hyperparameters requires defining a hyperparameter space, which includes the set of hyperparameters and their respective ranges that are considered to train a machine learning model (Weerts et al., 2020). This space is multidimensional, with each dimension corresponding to a different hyperparameter.

In this text, the search for the best hyperparameters was conducted in two different ways, Grid Search and Random Search. Both were validated using cross-validation. In a Grid Search, every possible combination of hyperparameters from a specified set is systematic explored. For each combination, the model undergoes training and evaluation.

The combination that yields the highest model performance is selected as the optimal configuration (Liashchynskyi and Liashchynskyi, 2019).

The grid search approach can demand significant computational effort, depending on the size of the hyperparameter space. This method is restricted by the range of hyperparameter values predefined, which might not encompass the best possible options. Grid search is a useful technique, especially for smaller and less complex models.

Random search involves selecting some combinations from the hyperparameter space at random to train and evaluate the machine learning model. This approach is useful when dealing with a large number of possible combinations. However, this method is less systematic and effective in identifying the optimal hyperparameter set (Bergstra and Bengio, 2012).

2.4 Threshold Tuning

In general, for binary classification, the threshold of classifier algorithm employs a default value of 0.5. This implies that if the probability of an instance is ≥ 0.5 to be positive class, it is assigned as positive class; otherwise, it is classified as negative class. Hence, for imbalanced dataset use 0.5 as threshold value is generally inappropriate (Zou et al., 2016).

The Receiver operating characteristic (ROC) curve plots True Positive Rate (hereafter named recall) versus False Positive Rate for different threshold probability from 0 to 1. True Positive Rate is the ratio of positive instances are correctly predicted, while False Positive Rate represents the ratio of negative instances are incorrectly predicted as positive. The ROC curve is a frequently employed tool for determining the optimal threshold value (Géron, 2022).

Figure 11 illustrate an example of ROC curve for different class distributions. In a perfect model (red curve), it will always be predicted 100% correct the positive class regardless of the threshold parameter, so the area under this curve (AUC) is 1. The AUC value tells how much the model is capable of distinguishing between classes. The green curve shows an AUC of 0.77, suggesting that the model has a 77% chance of accurately distinguishing between positive and negative classes. The dashed reference line indicates an AUC of 0.5, which means the model's performance is equivalent to random guessing, offering no real predictive advantage. On the other hand, the blue curve illustrates a model with an AUC of 0.23, indicating poor performance where most positive instances are incorrectly classified as negative.

Figure 12 shows the distribution curve of the positive (orange) and negative (blue) class in four different situations. The first case is the figure 12a, where the model is

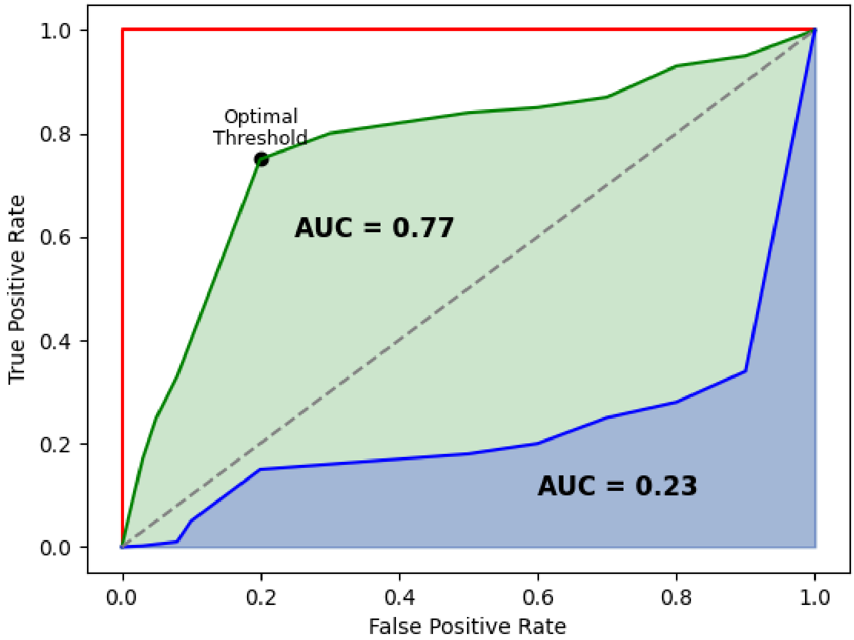


Figure 11 – The ROC curve representation of three hypothetical models. The red curve behaves like an ideal model, while the green and blue curves represent good and bad classifiers, respectively. The dashed line is the reference line.

perfectly able to distinguish between positive and negative class, in the reason that both distribution curves not overlap.

Figure 12b the distribution of the classes overlap, which introduce errors in classification task. This example is represented by green curve in figure 11, with a AUC values of 0.77, which mean that the model has 77% change to correctly distinguish the classes.

The figure 12c the distribution of classes completely overlap, in this case the AUC value is 0.5 and ROC curve of the model is the same of reference line (dashed black curve of figure 11). In the last example is the figure 12d, which has AUC of 0.23 represent the blue curve in the figure 11, in this case the model poorly classify the instances.

Good predictors tend to have an AUC value close to 1, and the optimal threshold probability can be determined by the value where the distance of ROC curve is closet to the point (0,1).

The outcome of machine learning predictions is divided in four categories:

- **True Positive (TP):** The outcomes that are correct predicted for positive class.
- **True Negative (TN):** The outcomes that are correct predicted for negative class.
- **False Positive (FP):** The outcomes that are incorrect predicted for positive class.
- **False Negative (FN):** The outcomes that are incorrect predicted for negative class.

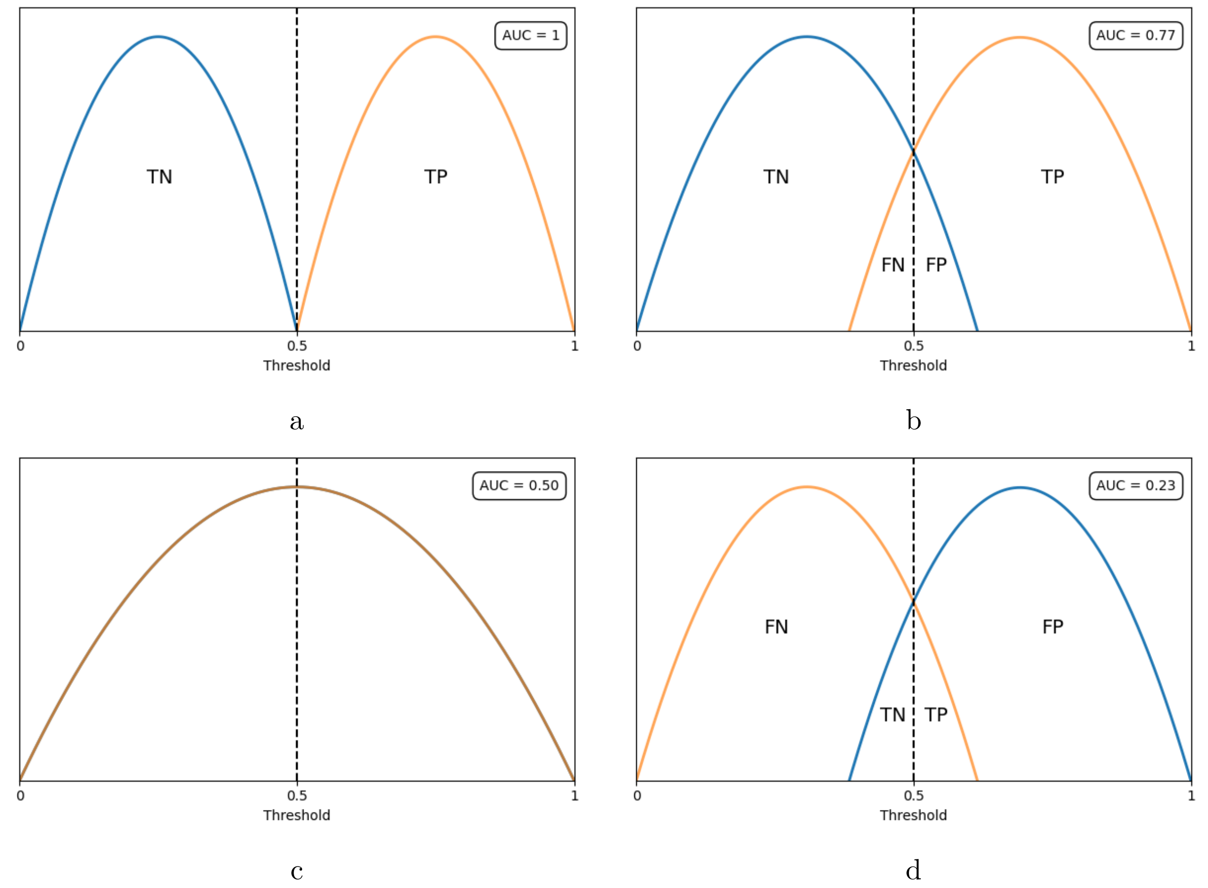


Figure 12 – The distribution of positive (orange) and negative (blue) class for different hypotetical models.

2.5 Performance evaluation

The performance of a machine learning model can be evaluated by comparing its predictions with a test dataset.

An important tool for visualizing the quality of a classification model is the confusion matrix. This statistical table maps the model's prediction results, showing the number of correctly and incorrectly predicted instances (Stehman, 1997). Figure 13 shows an example of a confusion matrix for binary classification.

The confusion matrix displays the actual number of instances for each class in its rows, and the predicted number of instances for each class in its columns.

It is divided into four categories: TP, TF, FP, and FN, and these categories are used to calculate the following metrics: accuracy, precision, recall, specificity, and F1-score.

The accuracy measures the rate of instances that the model correctly predicts, given by

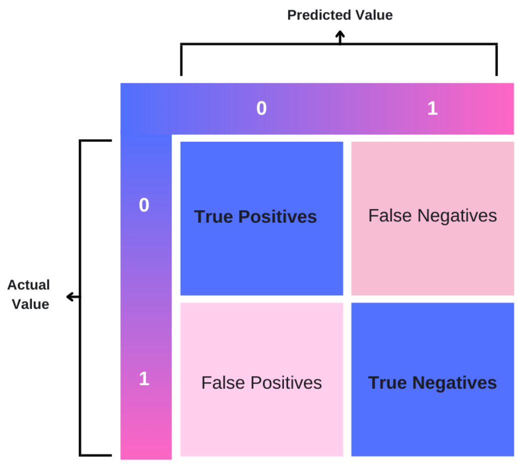


Figure 13 – Confusion Matrix for binary classification.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.9}$$

this metric may not be appropriate for imbalanced datasets, as it overlooks the class distribution. Other metrics can measure the model performance for a specific class.

Recall or the true positive rate is the ratio of correctly classified positive instances to the total number of actual positive instances:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.10}$$

Specificity or true negative rate is the ratio of correctly predicted instances within the actual negative class:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{2.11}$$

and the False Positive rate is given by $1 - \text{Specificity}$

Precision is the ratio of actual positive instances within the predicted positive class:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.12}$$

The F1-score is a way to measure the relation between the precision and recall in a single number. A significantly lower value for either precision or recall indicates that the model is not balancing these two metrics well when we want to give them equal importance. The F1-score combines precision and recall by calculating their harmonic mean:

$$\text{F1-score} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{2.13}$$

3 Machine learning approach for mapping stability regions around planetary systems

The number of confirmed exoplanets increases every year, leading to research into the stability of extrasolar planetary systems. To investigate potential regions for satellites or ring systems around these planets, researchers often use N-body numerical simulations, exploring a grid of orbital parameters as initial conditions.

Hunter (1967) studied the stability of satellites orbiting Jupiter under the influence of the solar gravity. Their results show that stable regions for prograde motion is 0.45 of the Hill radius ($r_{\rm Hill}$), while for retrograde motion, it extends up to 0.75 $r_{\rm Hill}$. In another study, Neto and Winter (2001) investigated gravitational capture within the Sun-Uranus-satellite system. They used a diagram of initial conditions, plotting the semi-major axis versus the eccentricity of particles, to identify stable regions around Uranus. Additionally, they considered particles with differents initial orbital inclinations and arguments of pericenter.

Holman and Wiegert (1999) conducted a numerical investigation into the stability of S-type orbit around a planet in elliptic three-body problem, consisting of star, planet, and particle. They analysed a range of mass ratio of μ = [0-0.9] and planet eccentricities of $e_{\rm p}$= [0-0.8]. Their results produced an empirical expression for the critical semi-major axis of stable particles in units of planet semi-major axis, initially in circular and planar orbit around the planet.

The three main types of orbits considered in binary or multi- body systems are: S-type orbits, which occur when a small body orbits one of the massive bodies in the system; P-type orbits, where a small body orbits around the barycenter of the system; and T-type orbits, which resemble the trajectories of Trojan asteroids and occur when the small body orbits near the triangular Lagrangian points L_4 or L_5 (Dvorak, 1986; Murray and Dermott, 1999; Holman and Wiegert, 1999).

Domingos et al. (2006) studied stability of hypothetical satellites orbiting extrasolar planets with a mass ratio of $\mu = 10^{-3}$ and close to their star (0.1 au). They derived an empirical expression for the boundary of stable regions in terms of Hill radius, eccentricities of both the planet and the satellite for satellites in prograde and retrograde orbits.

Rieder and Kenworthy (2016) performed N-body simulations to analyze the stability of a potential ring system around the candidate planet J1407b. Similarly, Pinheiro and Sfair (2021) conducted several numerical simulations of the three-body problem for a possible planet candidate (PDS110b) to host a ring system. Both papers refined parameters that remain undefined from observational data, such as the planet's mass and eccentricity, as well as the ring's inclination and size.

These works rely on numerical simulations that often incur substantial computational effort. A new approach to reduce the time of this process is to use N-body simulations to generate a dataset for training machine learning models, which, once trained, can predict outcomes for unseen data, significantly reducing computation time.

Tamayo et al. (2016) applied machine learning algorithms to predict the stability of tightly packed planetary systems with two planets orbiting a central star, achieving a predictive accuracy of 90% over the test set. Their findings reveal that this innovative approach is three orders of magnitude faster than direct N-body simulations. Another paper, Lam and Kipping (2018) employed deep neural networks (DNNs) to predict the stability of initially coplanar, circular P-type orbits for circumbinary planets, achieving an accuracy $\geq 86\%$ on the test set.

We propose an application of machine learning techniques to predict a stability map around hypothetical planet. We conducted numerous dimensionless numerical simulations of an elliptical three-body problem, consisting of star, planet, and test particle orbiting the planet. In contrast to Domingos et al. (2006), each simulation is characterized by nine initial parameters, including the orbital elements of the particle and the planet, as well as the mass ratio of the star-planet system.

Addressing this problem purely through numerical simulations could become computationally intensive, particularly as the number of grid parameters to be examined grows. While a single instance of the three-body problem may be computed relatively quickly, exploring a wider range of initial conditions can substantially raise computational demands, thus emphasizing the benefits of our machine learning approach.

This chapter is organized as follows: Section 3.1 describes the numerical simulations used to generate our dataset, the section 3.2 presents the machine learning models and their performance. Section 3.3 compares the predictions made by our best machine learning model with the results from Domingos et al. (2006) and Pinheiro and Sfair (2021), as well as with a stability map of Saturn's Inuit group of satellites. The final section provides final remarks.

3.1 Numerical simulations

Here, we explore the orbital stability of the three-body problem involving a star, planet, and test particle by modeling a set of dimensionless numerical simulations. Each simulation corresponds to a different initial condition for a hypothetical planetary system.

Initially, we define a range for the mass ratio of the system μ as

$$\mu = \frac{m_\mathrm{p}}{M_\mathrm{s} + m_\mathrm{p}} \tag{3.1}$$

where m_p and M_s represent the masses of the planet and the star, respectively.

A study involving 4,175 confirmed exoplanets from Schneider et al. (2011) shows that 4,150 of them (99.6%) have a mass ratio $\mu \leq 10^{-2}$. Consequently, we restrict μ to the range between 10^{-5} and 10^{-2} in our numerical simulations.

In all simulations, the masses of the star (M_s) and planet (m_p) are set as:

$$M_\mathrm{s} = 1 - \mu \tag{3.2}$$

$$m_\mathrm{p} = \mu \tag{3.3}$$

The semi-major axis of the planet a_p= 1, and its collision radius r_p = 5% of its Hill radius computed at the pericentre:

$$r_c = 0.05(1 - e_\mathrm{p})\sqrt[3]{\frac{\mu}{3}} \tag{3.4}$$

representing 55.6% of planet radius from the survey of exoplanet, as shown in the histogram in figure 14.

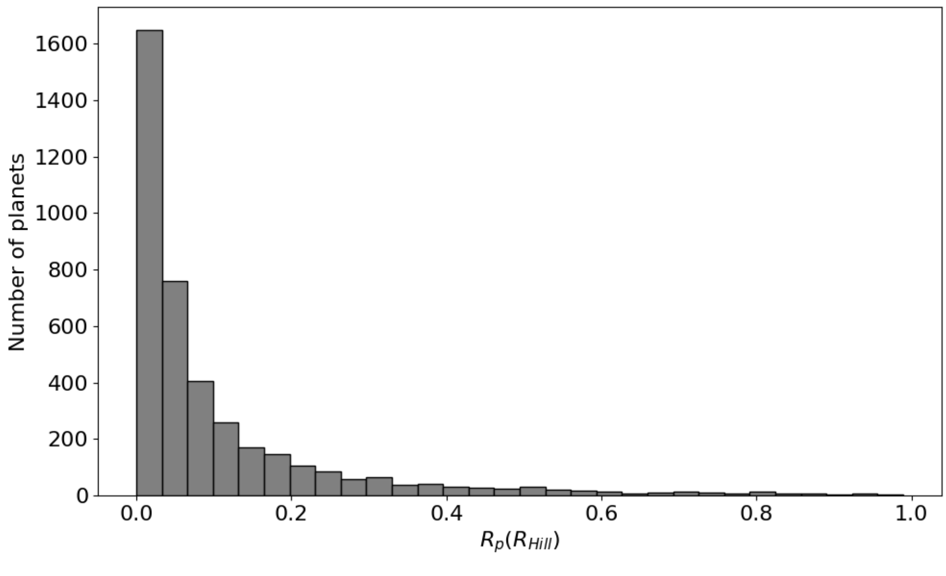


Figure 14 – Histogram of confirmed planets in units of their Hill radius from a survey consisting 4,150 different planets (Schneider et al., 2011).

The planet's eccentricity (e_p) and true anomaly (f_p) were randomly selected in an uniform distribution within the ranges: e_p= [0 - 0.99] and f_p= [0° - 360°]. All the other planet orbital elements are set to zero.

The test particle was in S-type orbit around the planet, influenced gravitationally by the star. We select from random uniform distribution its initial orbital conditions: the semi-major axis a_{p}= [1.1 r_{p}- 1 r_{Hill}], the eccentricity e = [0 - 0.99], the inclination i = [0° - 180°], and the argument of pericenter (ω), longitude of the ascending node (Ω), and true anomaly (f) all varied between 0° and 360°.

We conducted numerical integration using the REBOUND package and the IAS15 integrator (Rein and Spiegel, 2015), over a timespan of 10^4 orbital periods of the planet. The integration stops everytime when the particle collide with the planet reach an hyperbolic orbit (e $\geq$ 1) or it semi-major axis become a $>$ 1.

3.1.1 Numerical Results

In total, we run 10^5 numerical simulations, and the outcomes were as follows: 11.83% of the particles remained stable throughout the entire time of the integration, while 88.17% were unstable. Among the unstable particles, 47.22% collided with the planet, and 40.95% were ejected from the system.

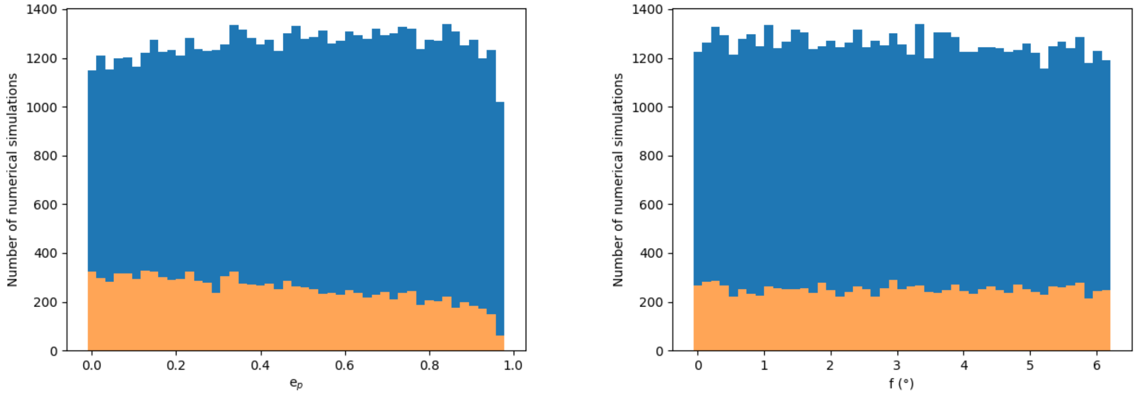
The results of the numerical simulations were used to create a dataset for training machine learning algorithms. This dataset included nine features, representing the initial conditions of each numerical simulation: the mass ratio of the system, the semi-major axis, inclination, argument of pericenter, longitude of the ascending node of the particle, as well as the eccentricity and true anomaly of both the planet and the particle. Each initial condition was labeled according to the numerical results in two different class: stable or unstable particle.

Figure 15 presents the distribution of each class in relation to the planet's orbital elements, while the figure 16 shows the distribution of stable and unstable classes for each feature related to the orbital elements of the test particles.

Four features demonstrate a strong correlation with the stability of the particle: planet eccentricity, particle semi-major axis, eccentricity, and inclination. The number of particles that collide increases as they are closer to the planet. The quantity of stable systems rises as the particles are initially positioned farther from the planet, peaking at 0.3 r_{Hill}. Beyond this point, stability decreases, with few systems remaining stable for semi-major axes a $>$ 0.7 r_{Hill}, where most of the particles are ejected.

The rate of unstable systems increases with higher eccentricity values. Eccentric orbits experience more significant gravitational disturbances at their pericenters, resulting in more collisions. No stable systems are observed when the initial particle eccentricity e $>$ 0.8.

Regarding particle inclination, retrograde orbits tend to contain more stable systems compared to prograde orbits, consistent with existing literature (Hunter (1967), Domingos et al. (2006)). No stable systems are observed for inclinations between 80° and 110°.



e_{p} f_{p}

Figure 15 – Histogram of the distribution of stable (orange) and unstable (blue) system along planet orbital elements

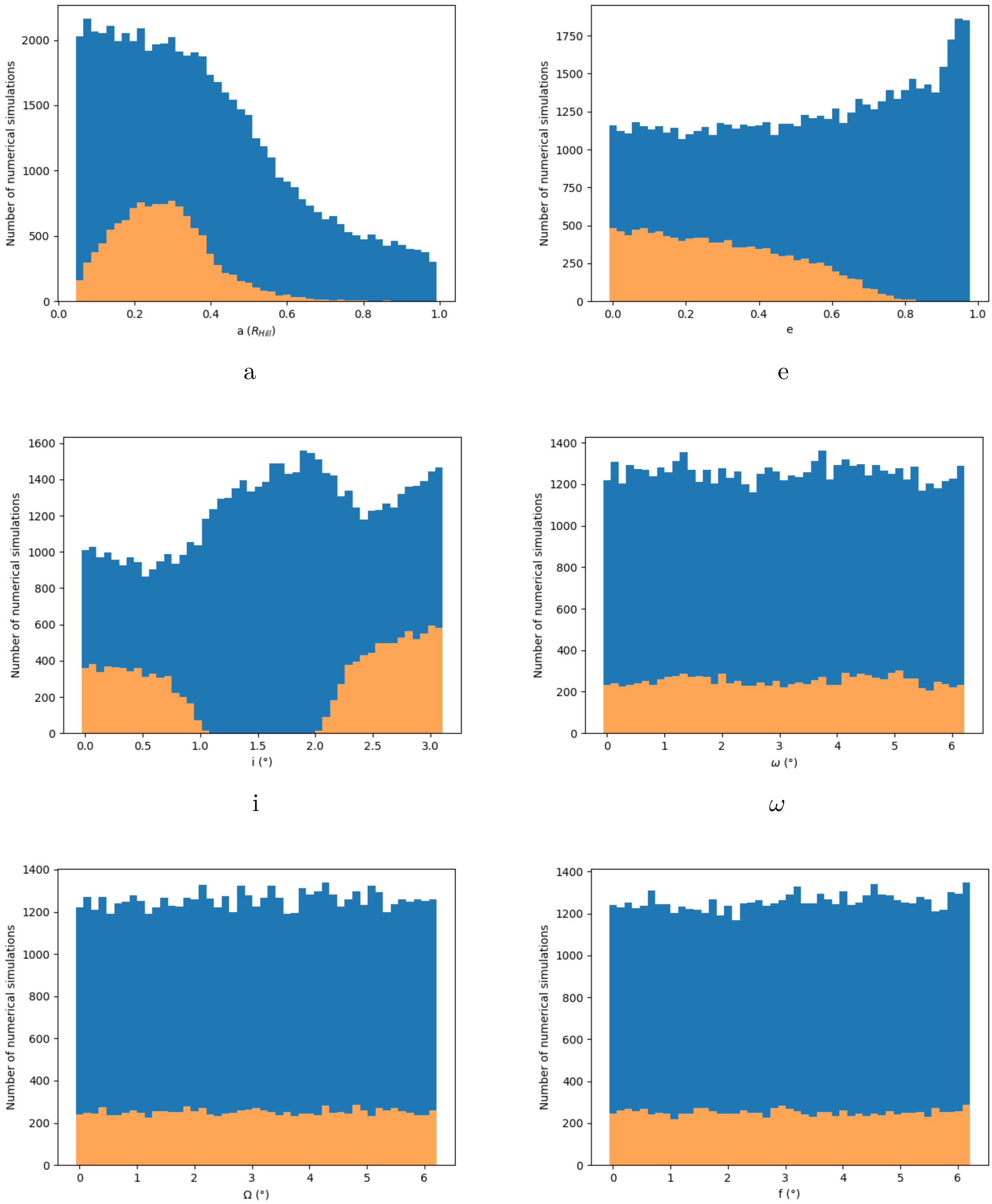
3.2 Performance Measures

We employed five different machine learning algorithms explained in chapter 2: Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost), Histogram Gradient Boosting, and Light Gradient Boosting Machine (LightGBM). We split our dataset, apportioning 80% for training and validation and 20% for testing the performance of the best machine learning model for each algorithm.

To address the issue of imbalanced class and enhance classifier performance, we used cross-validation to evaluate our models with and without various resampling techniques, including random undersampling, random oversampling, SMOTE, Borderline SMOTE, and ADASYN. Additionally, we conducted hyperparameter and threshold tuning to identify the best model for each machine learning algorithm.

This work is divided into five main steps, as shown in figure 17, which outlines the workflow. First step, we generated a dataset from numerical simulations. Second step, we tackled the problem of imbalanced classes using various resampling techniques. Third and fourth step, we performed hyperparameter tuning and adjusted threshold values to determine the optimal model configuration. In the final step, we evaluated the performance of the best model.

The Receiver operating characteristic (ROC) curve is a plot that shows how well a machine learning model performs across various classification thresholds, in the figure 18 shows the ROC curves for the best performing models of each algorithm.



a e

i ω

Ω f

Figure 16 – Histogram of the distribution of stable (orange) and unstable (blue) system along the orbital elements of the particle.

The area under the ROC curve (AUC) quantifies the overall ability of the model to discriminate between positive and negative classes, with an ideal model achieving an AUC of 1. In the table 1 shows the AUC values for the five algorithms.

All tested machine learning models exhibited performance close to the ideal ROC

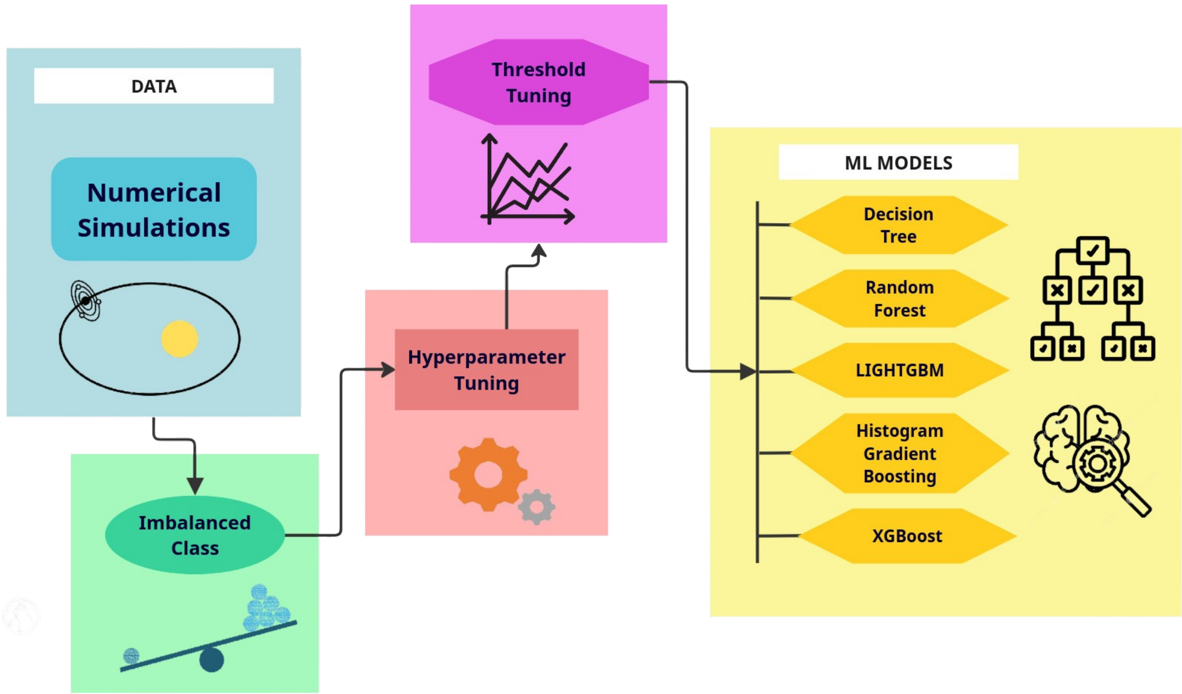


Figure 17 – Flowchart showing the stages of this project.

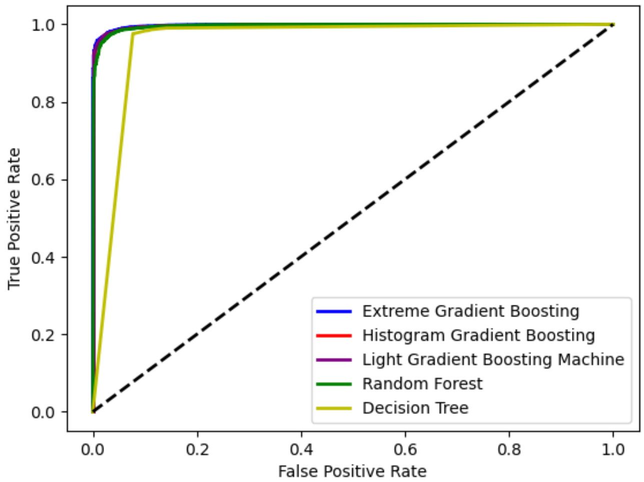


Figure 18 – The ROC curve of the best model of each algorithm.

curve. The Extreme Gradient Boost (XGBoost) model achieved the highest AUC value of 0.9978, and the Decision Tree model recorded the lowest AUC value of 0.9554, a difference between their AUC values of 0.0424

Table 2 presents the hyperparameters and threshold values associated with the best-performing model for each algorithm.

Model	AUC value
Extreme Gradient Boost	0.9978
Light Gradient boosting model	0.9975
Histogram Gradient Boosting	0.9974
Random Forest	0.9957
Decision Tree	0.9554

Table 1 – Area under the ROC Curve (AUC) for the best model of each algorithm.

	DT	RF	XGB	LGB	HGB
max_depth	None	None	8	None	None
min_samples_split	4	3	-	-	-
min_samples_leaf	3	2	-	20	20
max_features	None	None	-	-	1
n_estimators	-	200	-	-	-
booster	-	-	gbtree	gbdt	-
max_iter	-	-	-	-	200
learning_rate (eta)	-	-	0.2	0.2	0.2
scale_pos_weight	-	-	0.4	-	-
Threshold value	0.1	0.4	0.4	0.45	0.35

Table 2 – The hyperparameters and threshold values of the best-performing model for each algorithm. A trace line indicates that the algorithm does not have the specified hyperparameter. The algorithms are abbreviated as follows: Decision Tree (DT), Random Forest (RF), Histogram Gradient Boosting (HGB), Light Gradient Boosting Machine (LGBM), and Extreme Gradient Boosting (XGB)

The hyperparameters showed in the table 2 are:

- **max_depth**: The maximum depth of the tree. If set to None, nodes will continue expanding until either all leaves are pure, or until each leaf contains fewer than the minimum number of samples required for a split. (**min_samples_split**).
- **min_samples_split**: The minimum number of instances required to split an decision node.
- **min_samples_leaf**: The minimum number of instances required to be at a leaf node.
- **max_features**: The number of features to consider when looking for the best split.
- **n_estimators**: The number of trees in the forest.
- **booster**: Which booster the algorithm use.
- **max_iter**: The number of boosting iterations.
- **learning_rate (eta)**: It determines how large or small the steps are when the model adjusts its parameters during optimization to reduce the error or loss function. A lower learning rate results in smaller parameter updates, leading to slower but

more precise convergence. On the other hand, a higher learning rate allows for faster adjustments, potentially speeding up training, but it also increases the risk of overshooting the ideal solution.

- **scale_pos_weight** : Control the balance of positive and negative weights, useful for unbalanced classes.

Figure 19 shows the performance of five algorithms across four different metrics. The upper left panel presents accuracy, while the upper right panel is the precision for each class. The lower panels show recall (left) and F1-scores (right). Orange bars represent the stable class, while blue bars indicate the unstable class.

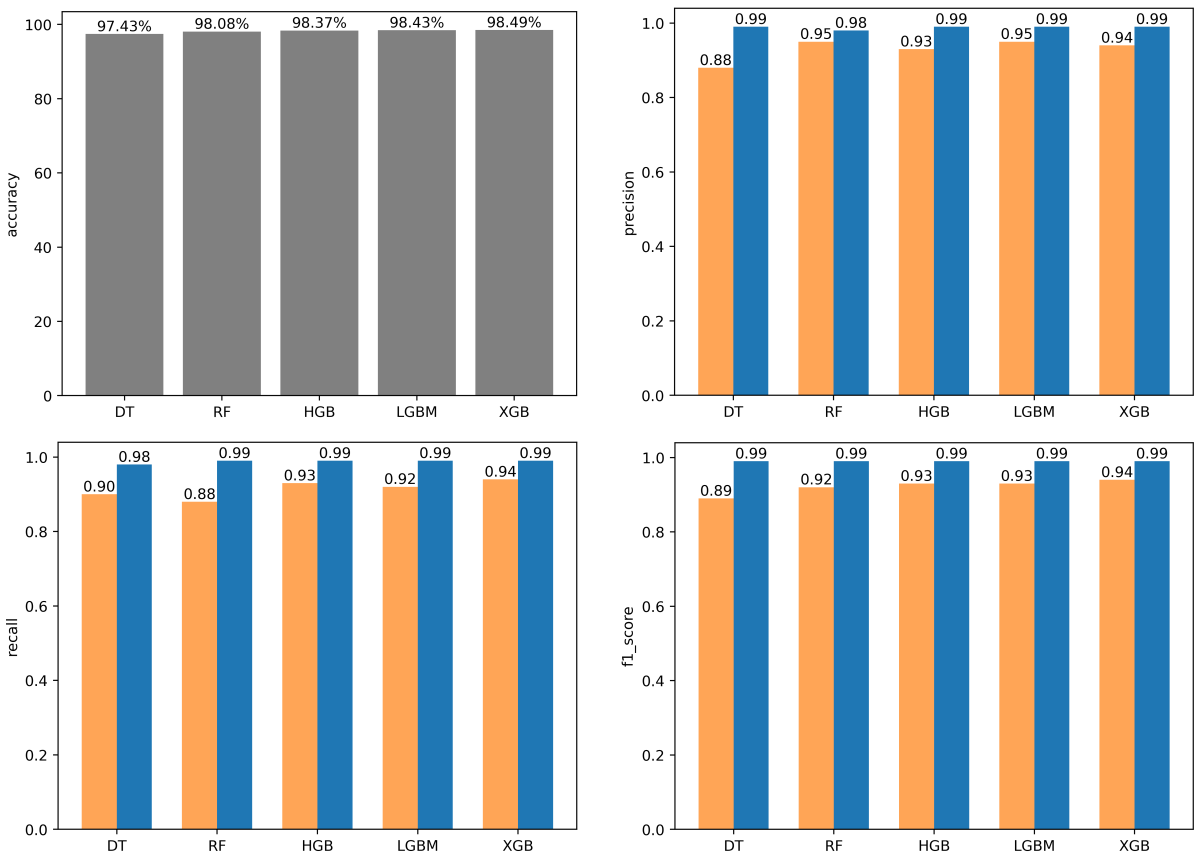


Figure 19 – The performance of the best model of each machine learning algorithm. In the upper panels, showing accuracy and precision, meanwhile the lower panels are recall and F1-score. The stable and unstable classes are represented by orange and blue bars, respectively.

All machine learning models achieved comparable accuracy performance, more than 97%, the difference in accuracy between the highest (XGBoost) and lowest (Decision Tree) values is 1.05%. An important consideration is that accuracy measures the percentage of correctly classified particles across all predictions. However, this positive result does not ensure good performance for each class, especially in the context of an imbalanced dataset where unstable systems outnumber stable ones by a factor of 7.5. In such cases, it is also important to consider other metrics, such as recall, precision, and F1-score for each class.

The recall and precision of unstable class consistently reached a high percentage of 99%, except for Random Forest, which achieved a precision of 98%, and Decision Tree that scored 98% for recall. In terms of performance on the stable class, both LightGBM and Random Forest achieve the highest precision values, reaching 95 % in their predictions, although their recall score were 92% and 88%, respectively. XGBoost achieves identical precision and recall percentages, resulting in the highest F1-score of 94%. In comparison, LightGBM and Random Forest achieve F1-scores of 93% and 92%, respectively.

3.2.1 Feature importance

The feature importance measures how effective each feature is in reducing impurity across all splits in the decision trees. This importance is calculated by averaging the contribution of each feature over all the decision trees, with the sum of all attributes being 1. The importance of each feature x_i is obtained by summing its contribution across all splits where that feature is applied:

$$\text{Importance}(x_i) = \frac{1}{T} \sum_{\text{tree}=1}^{\text{T}} \Delta\text{Impurity}(\text{tree}) \tag{3.5}$$

where T is the number of tree and ΔImpurity(tree) is reduction impurity, given by:

$$\Delta\text{Impurity}(\text{tree}) = \text{G}_{\text{Decision node}} - (\frac{\text{m}_{\text{left}}}{\text{m}}\text{G}_{\text{left}} + \frac{\text{m}_{\text{right}}}{\text{m}}\text{G}_{\text{right}}) \tag{3.6}$$

where m is the number of instances and G is gini impurity score, it is also possible to use the entropy (equation 2.4) instead gini impurity. Figure 20 shows the relative importance of the features in our classification task.

The top three features are particle orbital elements: semi-major axis (a), eccentricity (e), and inclination (i), with importance scores of 0.3567, 0.2783, and 0.2351, respectively. These results align with the distribution patterns discussed in subsection 3.1.1, where we observed significant reductions in stable systems beyond 0.6 $r_{\text{Hill}}$and the absence of stable particles with inclinations close to 90° or eccentricities ≥ 0.8 (figure 16).

The planet's eccentricity is the fourth most influential attribute, with a value of 0.0564 followed by the particle's argument of pericenter at 0.0274. As shown in Figure 15, there is a decrease in the quantity of stable systems as the planet's eccentricity increases. The remaining parameters particle true anomaly (f) and longitude of node (Ω), mass ratio (μ), planet true anomaly (f_p) have little impact on the classification, with importance values below 0.012.

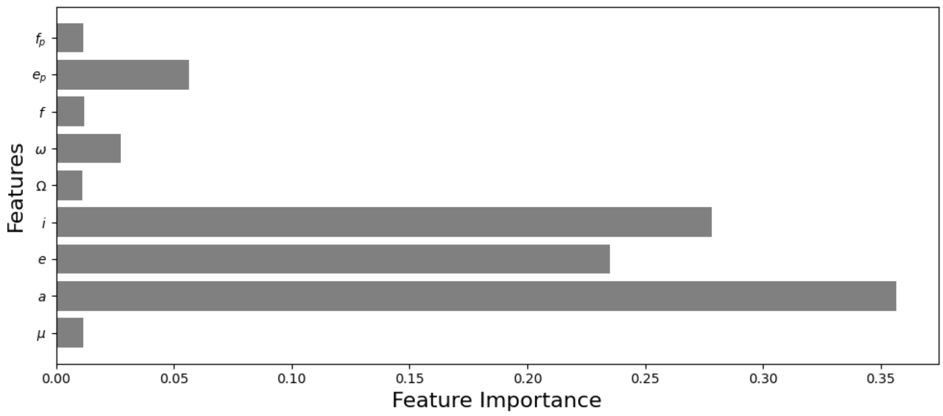


Figure 20 – Feature importance derived from the XGBoost model. Orbital elements are denoted as follows: semi-major axis (a), eccentricity (e), inclination (i), argument of pericenter (ω), longitude of node (Ω), and true anomaly (f). Planet-specific parameters are indicated with a 'p' subscript. The mass ratio of the system is represented by μ.

The semimajor axis is related to the average distance of the particle from the planet. When a particle orbits near a planet, the planet's stronger gravitational pull increases the likelihood of orbital instabilities. Farther from the planet, the star's gravitational influence dominates, which can also destabilize the particle by pulling it away. Eccentric orbits can also influence the stability of the particle, as the particle experiences stronger gravitational perturbations near the pericenter, where it is closest to the planet. The particle's inclination relate to the planet's orbit, also plays a significant role. Retrograde orbits tend to be more stable than prograde ones, as demonstrated in the studies by Domingos et al. (2006) and Pinheiro and Sfair (2021).

3.3 Predicting stable maps in planetary system

We apply our best machine learning model to predict four distinct stability maps, aiming to replicate the results of Domingos et al. (2006) (subsection 3.3.1), Pinheiro and Sfair (2021) (subsection 3.3.2), and a specific group of Saturn's moons known as Inuit (subsection 3.3.3). Each diagram of initial conditions consists of 10,200 data points, where test particles are uniformly distributed within an initial semi-major axis range from 1.1 $r_{\rm p}$to 1 $r_{\rm Hill}$, with a step size of Δa = 0.1 $r_{\rm p}$, and an initial eccentricity e = [0 - 0.5], with Δe = 0.005.

3.3.1 Stability around extrasolar giant planets

Domingos et al. (2006) numerically simulated the elliptic three-body problem involving a star, a planet, and a satellite. The star-planet system have a mass ratio of $\mu = 10^{-3}$. The planet is located close to its star, with a semi-major axis of 0.1 au and eccentricity varying in range of e_p = [0 - 0.9] in step of $\Delta e_p = 0.1$. The test particles were uniformly distributed in semi-major axis and eccentricity, generating a diagram of initial conditions for each planet parameter. The authors derived an analytical expression for the critical satellite semi-major axis as a function of the eccentricity of both the satellite and the planet. The authors explored the case where the particle's inclinations 0° (prograde orbit) and 180° (retrograde orbit).

To compare the predictions of our machine learning model with numerical results, we numerically reproduce two different stability maps explored in Domingos et al. (2006). Both cases involve a system with $\mu = 10^{-3}$, a planet in a circular orbit, and radius of 5% of its Hill radius. In the first scenario, a prograde particles with orbital inclination of i = 0°, while the second one a retrograde particles with i = 180°.

In the figure 21 shows the performance over the test set of machine learning predictions for both case, in dashed lines for prograde particle and solid lines for retrograde particles.

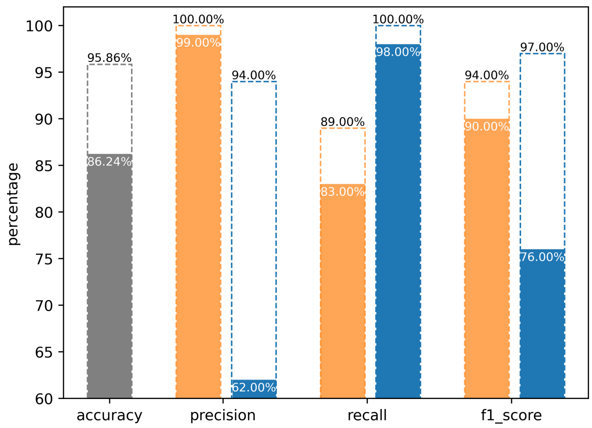


Figure 21 – Performance comparison between numerical simulations and machine learning predictions for prograde (dashed line) and retrograde (solid line) orbits from Domingos et al. (2006). Blue and orange regions represent unstable and stable particles, respectively.

The figure 22 shows the outcomes of numerical simulation (upper panels) and machine learning (lower panels). In the left panels illustrate the case of prograde particles, while the right panels are retrograde particles. The dashed lines represent the analytical expression derived by Domingos et al. (2006).

In both cases, the machine learning model successfully reproduces the main structure of the stable region, and misclassified initial conditions on boundaries. For prograde orbits, the model correctly predicts $\sim$ 96% of the data points, while in the case of retrograde orbits achieved an accuracy of $\sim$ 86%.

The prograde region extends up to approximately 0.45 r_{Hill}, so the stable region predicted by our machine learning model is slightly smaller. At the edges of this region, the model incorrectly classifies some stable particles as unstable. Despite this, nearly all particles identified as stable by our model are indeed stable, with only 5 out of 3,447 particles being misclassified, resulting in a precision of approximately 100%. Additionally, all particles that are actually unstable were correctly classified.

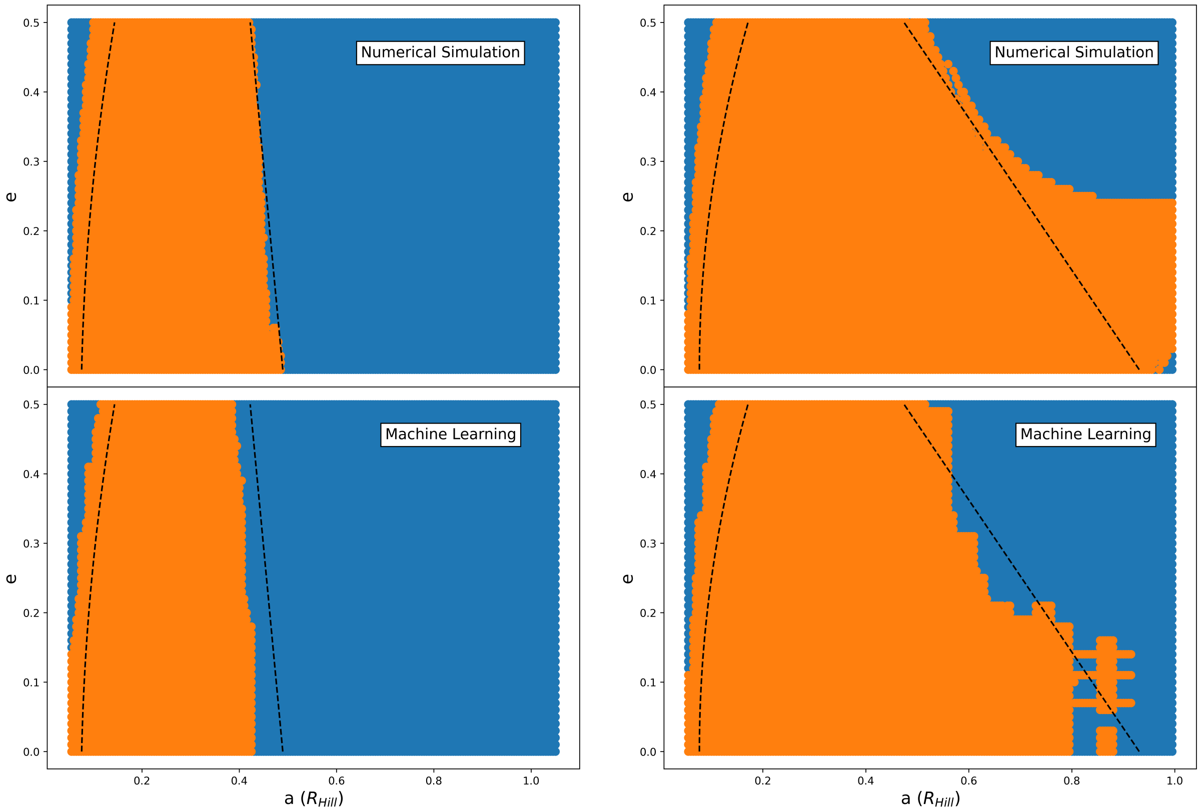


Figure 22 – Comparison of numerical simulations (upper panels) and ML predicted outcomes (lower panels) for stable (orange) and unstable (blue) particles. Initial conditions match those from Domingos et al. (2006) for $\mu = 10^{-3}$, with i $= 0°$ (left) and i $=$ 180° (right).

The comparison between the analytical formula for the critical semi-major axis introduced by Domingos et al. (2006) and our machine learning model demonstrates comparable overall performance. While the equations proposed by Domingos et al. (2006) exhibit slightly higher accuracy (96.05%) and recall for the stable and unstable categories (91.80% and 98.70%, respectively), our machine learning model surpasses in precision for both classes. Specifically, the method by Domingos et al. (2006) achieves a precision of 97.78% for stable particles and 95.08% for unstable particles, whereas our machine learning model attains 100% precision for both classes. The F1-score for both approaches remains identical.

The stability map for retrograde particles shows approximately twice as many stable particles as the map for prograde particles. In this case, the stable region extends up to 1 r_{Hill}, while the machine learning predictions extend to about 0.9 r_{Hill}. Our model misclassified many stable particles located beyond 0.8 $r_{\text{Hill}}$with eccentricities < 0.3, as well as particles at the edge of the stable region. Nevertheless, 99% of the particles classified as stable by the model are indeed stable, with only 44 out of 6,200 particles being misclassified. The model also correctly identified 98% of the unstable particles.

Despite these constraints, the machine learning model achieved an accuracy of 86.24%, surpassing the analytical formula proposed by Domingos et al. (2006), which had an accuracy of 85.04%. Additionally, the ML model achieved a precision of 99% for stable particles and 62% for unstable particles, along with a recall of 83% for stable particles and 98% for unstable particles. These metrics resulted in F1-scores of 90% for stable particles and 76% for unstable particles.

In contrast, the equations developed by Domingos et al. (2006) produced slightly lower F1-scores, achieving 89.26% for stable particles and 74.85% for unstable particles. The recall for the stable class was 80.81%, while the precision for the unstable class reached 59.94%. Nevertheless, their precision and recall for stable and unstable particles were marginally superior, attaining values of 99.83% and 99.54%, respectively.

These results indicate that the model can accurately identify the overall structure of stable regions for both prograde and retrograde orbits. However, the predictions of our machine learning model need improvement, particularly at the boundaries of stable regions, perhaps expanding the dataset might help address these inaccuracies.

3.3.2 Stability of PDS110b ring system

The star PDS110 underwent two similar dimming event in 2008 and 2011 (Osborn et al., 2017). A possible explanation for both eclipse is a passage of inclined exoplanetary ring system in front of star. Pinheiro and Sfair (2021) conducted an order of 10^6 numerical simulations of the three-body problem, aiming to constraints orbital and physical parameters for this system.

We performed numerical simulations and compared the result of stability maps with machine learning predictions for the scenario in which the candidate exoplanet PDS110b has a mass of 10 Jupiter masses, an eccentricity of 0.2, and it is orbited by particles with an inclination of 150°.

In figure 23 the numerical results are shown on the left and the machine learning predictions on the right. The black line represents the radial size of the observed ring as reported by Pinheiro and Sfair (2021). In this case, the number of stable and unstable particles is roughly equivalent, with 5,044 stable and 4,599 unstable particles.

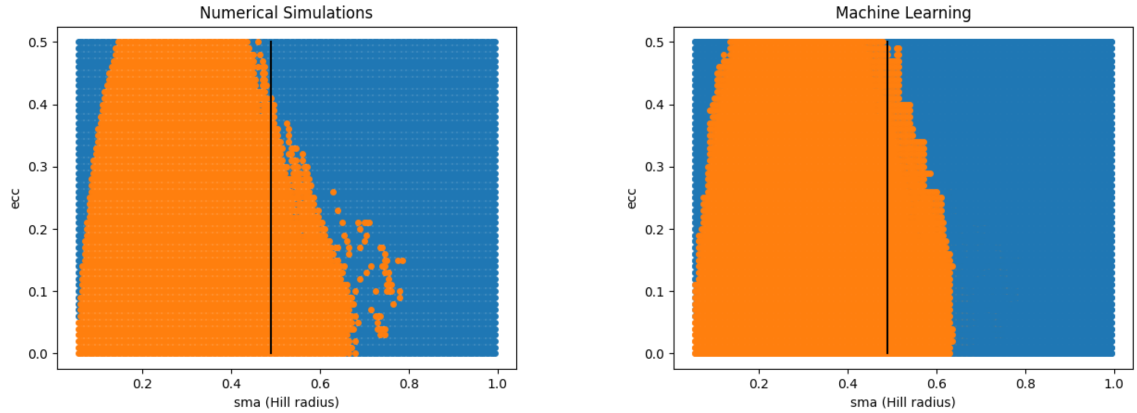


Figure 23 – Stability maps for PDS110b. Top panel: Numerical simulation results. Bottom panel: Machine learning prediction. The black line represents the minimum radial size for the observed ring as reported by Pinheiro and Sfair (2021). Blue and orange regions represent unstable and stable particles, respectively.

The machine learning model successfully reproduced the main structure of the stable region. However, similar to the previous examples, it misclassified particles at the boundaries, with 159 particles from the unstable set and 297 from the stable set being incorrectly classified. Some of these misclassified particles are outliers located outside the stable region. Our numerical simulations cover a time span of 10^4 orbital period, and it's possible that some stable particles might become unstable if integrated over longer periods.

Figure 24 shows the performance metrics, with the stable class represented in orange and the unstable class in blue. Across all metrics, the model achieved a percentage range between 94% and 97%, with the highest score being the recall for the stable class and the lowest values were the precision for the stable class and the recall for the unstable class.

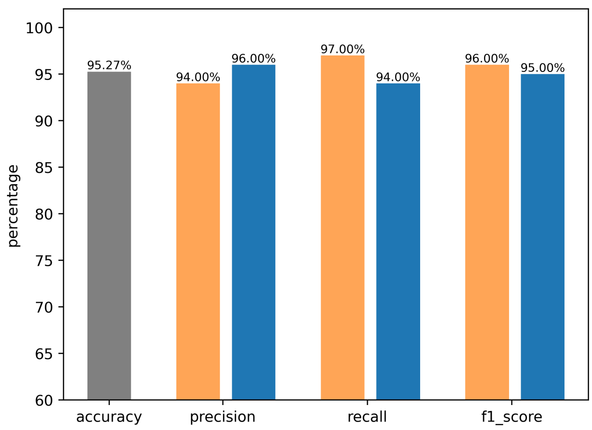


Figure 24 – Performance metrics for the PDS110b system prediction. Blue and orange regions represent unstable and stable clases, respectively.

3.3.3 Stable region around Saturn - Inuit group

Many of Saturn's irregular satellites can be classified into different groups according to their semi-major axis and inclination. These groups are the Inuit, Gallic, and Norse. Irregular satellites are generally eccentric objects and are believed to be remnants of the solar system formation that were captured by Saturn or collisional fragments (Ashton et al., 2021).

The Inuit group consists of prograde satellites that share similar characteristics, such as their distances from the planet, which range between 0.16 to 0.3 Saturn's Hill radius, and their orbital inclinations, which are approximately between 45° (Sheppard et al., 2023; Grav and Bauer, 2007). The orbital elements of the satellites belonging to this group are presented in Table 3.

The Gallic group is composed of prograde satellites within a distance range from Saturn of 0.24 to 0.35 Saturn's Hill radius and with orbital inclinations around 37°. The Norse group contains a large number of retrograde irregular satellites of Saturn, with inclinations $> 130°$ (Sheppard et al., 2023; Grav and Bauer, 2007).

Moon	a (r_{Hill})	e	i (°)
Kiviuq	0.1803	0.334	45.71
Ijiraq	0.1805	0.316	46.44
S/2019 S1	0.1825	0.623	44.38
Paaliaq	0.2466	0.364	45.13
Siarnaq	0.2844	0.295	45.56
Tarqeq	0.2922	0.160	46.09
S2428b	0.2835	0.472	44.43
T522499	0.2824	0.242	48.11

Table 3 – The orbital elements of Saturn satellites belongs to Inuit group. Some Inuit satellites that were recently discovered were not included in this table because their orbital eccentricities are not well determined (https://carnegiescience.edu/).

We numerically simulated several different initial conditions of the three-body problem, involving the Sun, Saturn, and a test particle. The test particle had an orbital inclination of 45°, representing the Inuit satellite group. All the others particle's orbital elements, the argument of pericenter, true anomaly, and longitude of node were randomly selected from 0° to 360° using a uniform distribution.

Figure 25 presents the numerical results and machine learning predictions for this case. The black square points represent the satellites belonging to the Inuit group. This example achieved the best overall performance among all the cases tested. Our model misclassified only 333 particles out of 10,200, resulting in an accuracy of 96.73% (Figure 26). The machine learning model correctly identified 93% of stable particles and 99% of unstable particles. As with the other examples, the machine learning model accurately reproduced the main structure of the stable region, though it incorrectly predicted some particles at the boundary of this region due to generalization process.

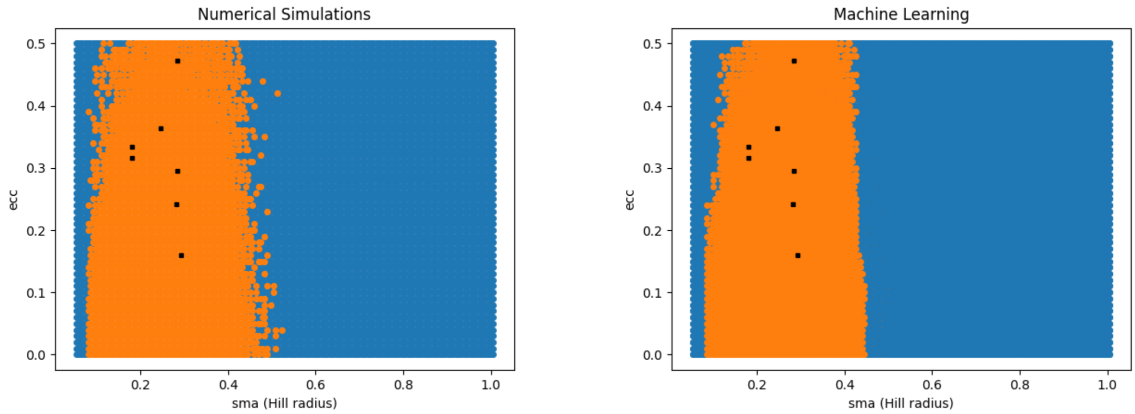


Figure 25 – Stability maps for Saturn's Inuit satellites. Top panel: Numerical simulation results. Bottom panel: Machine learning prediction. Black squares represent the actual satellites belonging to the Inuit group. Blue and orange regions represent unstable and stable particles, respectively.

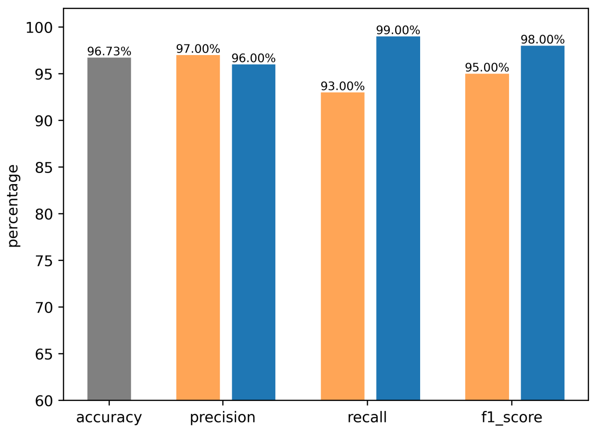


Figure 26 – Performance metrics for the Saturn's Inuit satellites prediction. Blue and orange regions represent unstable and stable clases, respectively.

3.4 Final remarks

This research highlights the effectiveness of machine learning in predicting the orbital stability of hypothetical planetary systems. We generate a dataset from 100,000 of the elliptic three-body problem, consisting of a star, a planet, and a test particle in an S-type orbit around the planet. The simulations consider a wide range of orbital and physical parameters, including the mass ratio of the star-planet system, the planet's eccentricity and true anomaly, as well as the test particle's semi-major axis, eccentricity, inclination, argument of pericenter, longitude of the node, and true anomaly.

Our simulations indicated that 11.83% of the test particles remained stable throughout the entire simulation, while 47.22% collided with the planet, and 40.95% were ejected from the system. Due to this imbalanced class distribution, we employed resampling techniques to improve model performance. We assessed five machine learning models: Random Forest, Decision Tree, XGBoost, Light Gradient Boosting, and Histogram Gradient Boosting. After thorough hyperparameter tuning and threshold optimization, XGBoost demonstrated the best performance, achieving an accuracy of 98.48%.

To validate the model's ability to generalize, we applied it to replicate the numerical results from studies by Domingos et al. (2006), Pinheiro and Sfair (2021), and a stability map for Saturn's Inuit group of satellites. The model consistently performed well, with a 96.73% of accuracy across these varied scenarios.

Our findings demonstrate the potential of machine learning algorithms to significantly reduce computational time. The machine learning model can handle a wide range of initial conditions in a matter of seconds, representing a notable improvement in efficiency over traditional numerical methods. Looking ahead, we are working on further enhancing the model's performance, particularly in detecting stability at boundaries, and expanding our dataset to improve the model's generalization. We plan to make our model available through a public web interface to benefit the broader scientific community.

4 Refining physical and orbital parameters for possible extrasolar planetary ring systems

In our Solar System, all four giant planets have ring systems. Over the last decade, three small bodies have been discovered to also host rings: the Centaur Chariklo (Braga-Ribas et al., 2014), and the dwarf planets Haumea (Ortiz et al., 2017) and Quaoar (Morgado et al., 2023). Nonetheless, no extrasolar ring system has been confirmed yet. Several studies have systematic searched for signatures of a exoplanetary ring system, such as those by Heising et al. (2015), Aizawa et al. (2017), and Aizawa et al. (2018).

Most exoplanets have been discovered using photometric transit techniques. Rings around these planets would cause additional dips in the light curve with a specific pattern in different transit. Unlike satellite, which produce dips at distinct positions relative to the planet across multiple transits, unless it is in orbital resonance with the star and the planetary orbit (Heller, 2017).

According to Barnes and Fortney (2004), the detection of a planet with Saturn-like rings during a transit requires high photometric precision, achievable only by space missions like the Kepler mission. Tusnski and Valio (2011) simulated planetary transits, considering the effects of Saturn like ring around the planet. They authors concluded that such a ring system would indeed be detectable using Kepler. However, using CoRoT, this ring system might be misinterpreted as a single planet with a slightly larger radius.

The presence of rings can also be detected in the spectroscopy of the star through the Rossiter-McLaughlin effects. If during the transit, the planet-ring system orbited the star in the same direction as its rotation, they would block part of the blue-shifted radial velocity of the star during half of the eclipse, and in the other half, part of the red-shifted velocity. However, if the planet and the ring are orbiting the star in the opposite direction of its rotation, they would first block the red-shifted velocity (Heller, 2017). Figure 27 from Ohta et al. (2008) shows a theoretical model explain of how change the shape of light curve and radial velocity during the passage of a planet with and without a ring.

The first exoring candidate was reported by Mamajek et al. (2012), in 2007 the star J1407 underwent a series of complex eclipses lasting ~ 56 days and reaching a dimming of $> 95\%$. One possible explanation for this event is the passage of a giant ring extending up to a radius of 0.6 au around an unseen secondary companion. The eclipse structure exhibits that this ring may have a gap at 0.4 au, possibly caused by an hypothetical exosatellite with a mass comparable to Earth that cleared out this region (Kenworthy et al., 2015).

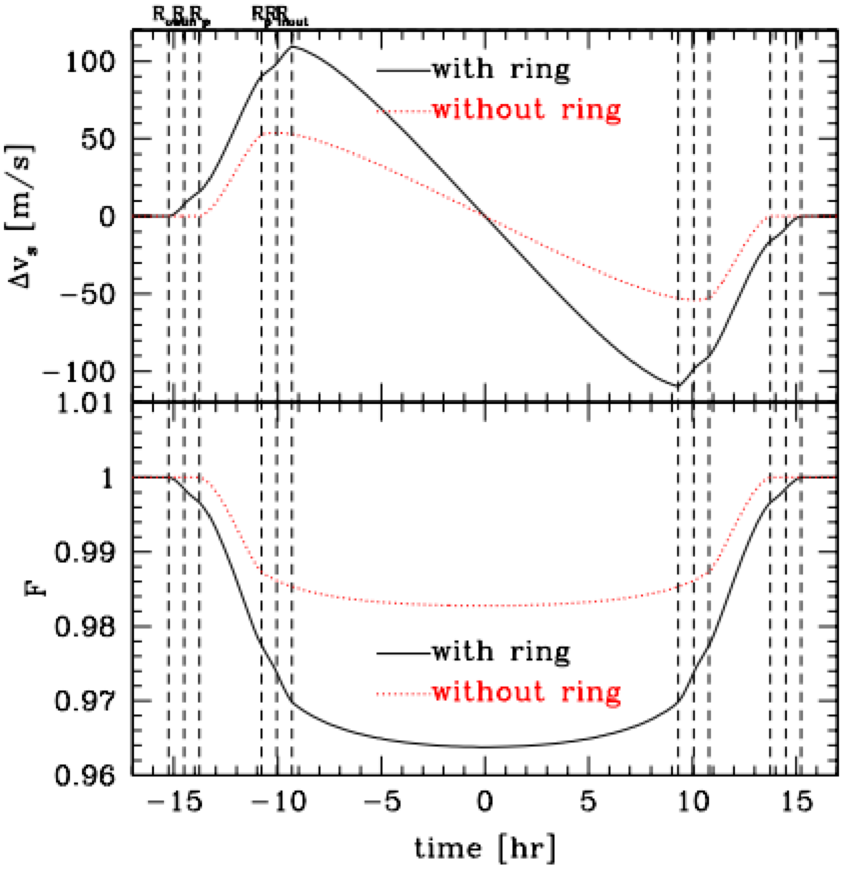


Figure 27 – The effect during the transit of the planet with and without a ring on the radial velocity (upper figure) and on the light curve (lower figure) of the star, through ground-based observations. In this scenario, the hypothetical planet orbits a Sun-like star at 3 au from star with size of 1.347 Jupiter's radius, and hosting a ring system with radii equivalent to 2 planet's radius (Ohta et al., 2008).

Osborn et al. (2017) detected two similar eclipses underwent by star PDS110 in 2008 and 2011, each lasting about 25 days and causing a dimming of approximately 30%. The similarity between these eclipse events suggests the existence of a possible secondary companion surrounded circum-secondary disc structure with radius of ~ 0.2 au. A ground-based observational campaign tried to observe a third eclipse predicted to happen in 2017. However, no significant deeps of $> 1\%$ was observed. Osborn et al. (2019) propose an alternative explanation, suggesting that rather than a ring structure, the variability in PDS110's luminosity could stem from a circumstellar dust disc. Further observations are needed to confirm the presence of this ring system.

This study aims to constraint the mass and eccentricity of the planets, as well as the size and inclination of the rings, assuming that these two eclipsing secondary discs are associated with giant planets (PDS110b and J1407b). Additionally, we estimate the likelihood of observing these dimming events for different line of sight. We utilized the machine learning model from chapter 3 to predict stable maps varying initial orbital and physical parameters of those systems. This allowed us to calculate hypothetical eclipse duration and identify those cases that matches with observed events.

The organizational structure of this chapter is as follows: Sections 4.1 and 4.2 introduce previous research and describe the ring systems of PDS110b and J1407b, respectively. Section 4.3 details the methodology employed in this study. Section 4.4 discusses the results and constraints on possible parameters for both observed ring systems. Lastly, in section 4.5 presents the final remarks and next steps.

4.1 The PDS110b ring system

The 11 million years old star PDS110, also known as HD290380, IRAS05209-0107, GLMP 91, 2MASS J05233100-0104237, and TYC 4753-1534-1, in the Orion OB1 association underwent to two dimming events in 2008 November and 2011 January, both having similar characteristics, with depth of $\sim 30\%$ and lasting ~ 25 days (Osborn et al., 2017).

The star radii and mass is equivalent to 2,23 $R_\odot$ and 1.6 $M_\odot$, and its effective temperature is estimated in 6,400 K. Additionally, it exhibits an infrared excess, suggesting the presence of circumstellar material. Osborn et al. (2017) results suggests the presence of two disks. The innermost disk has a radius of 0.2 au, which is too small to be directly observed through imaging techniques, while the outer disk is inclined and extends up to 300 au.

Osborn et al. (2017) interpreted those transits were caused by a passage of the edge of a circumsecondary disk, in two different scenarios. The first hypothesis suggests stellar disk around an unseen brown dwarf, implying that such a body has a low orbital inclination relative to the inner disk of primary star, and a inclined enough disk to pass through the line of sight of the primary star. Such scenario would likely be detected by the optical spectrum.

In the second hypothesis, a transit of giant planet (PDS110b) hosting a inclined ring system relative to its orbit. According to Osborn et al. (2017), this is the most plausible scenario, and the secondary body could be invisible during the eclipse. Figure 28 shows a schematic of the PDS110 system: the star at the center, surrounded by its inner dusty disk, orbited by the exoplanet PDS110b and its observed ring, and subsequently the second and more extensive and inclined circumstellar disk.

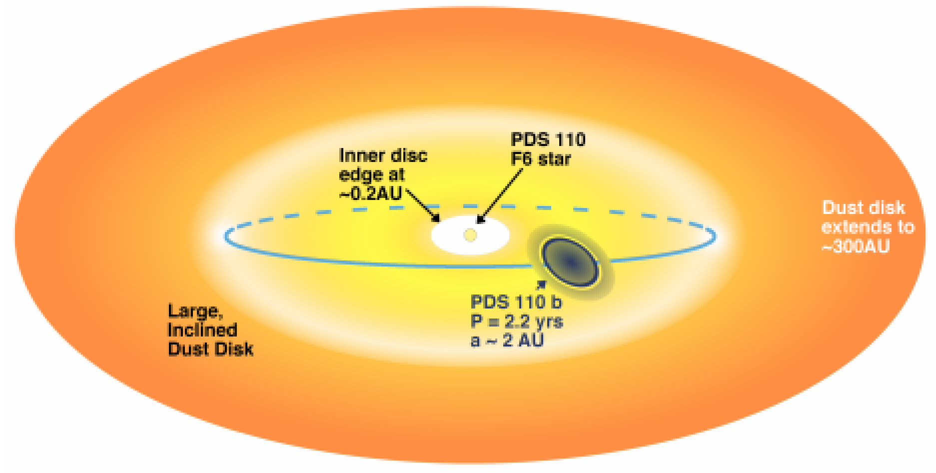


Figure 28 – A sketch of the PDS 110 system (Osborn et al., 2017).

In the upper panel in figure 29 shows the light curve data for these two different dimming events, the photometric data from 2008 (in yellow) and from 2011 (in blue), with day zero marking the center of each respective eclipse. The lower panel presents a schematic of the ring system, where the color intensity indicates the ring's transmission. The green band represents the path and diameter of the star PDS110 during the eclipse.

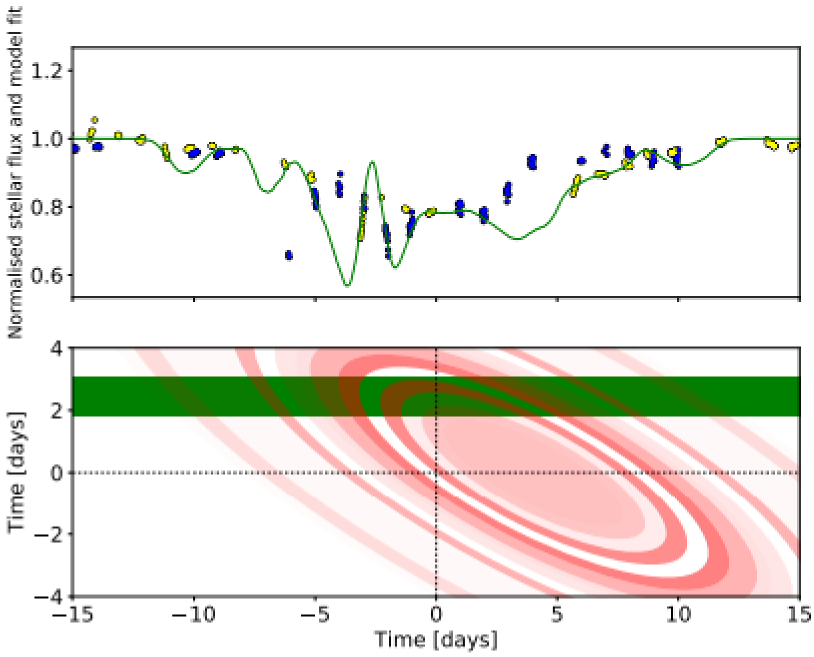


Figure 29 – The fit model for PDS110b's ring system. The upper panel shows the photometric for observed eclipses in 2008 and 2011, in yellow and blue colors, respectively. The lower panel provides a schematic representation of the ring system, illustrating the track of star (green) behind the rings. (Osborn et al., 2017).

Pinheiro and Sfair (2021) constraints on the mass and eccentricity of the planet, as well as the size and inclination of its ring system, through numerical simulations the authors explored a broad range of configurations for the PDS110b ring system and ruled out those that did not align with observations. Their results indicate that the ring system could be either prograde or retrograde, in the case of prograde ring, it inclination $< 60°$ with a radius between 0.1 and 0.2 au, the secondary companion is likely more massive than 35 M_{Jup} in a low eccentric orbit (< 0.05). For retrograde ring system, the eclipse can be explained by a secondary mass between 5 M_{Jup} to 70 M_{Jup}, with an eccentricity < 0.4.

4.2 The J1407b ring system

The star 1SWASP J140747.93-39454.6, also known as J1407, is a young star of 16 million years old, in distance of 113 parsecs from Earth and it belongs to the pre-main-sequence star, associated with the Scorpius–Centaurus association (Sco OB2). J1407 has an estimate mass of 0.9 $M_{\odot}$, with radius of 0.96 $R_{\odot}$, a temperature of 4400 K, and a rotational period of 3.21 days (Van Werkhoven et al. (2014), Kenworthy et al. (2015)).

A ground-based observational campaign using SuperWASP (Super Wide Angle Search for Planets) and ASAS (All Sky Automated Survey) telescopes, the star J1407 shows a peculiar light curve that covered a timespan of 56 days, from march to may 2007.

According to Van Werkhoven et al. (2014), only the star itself could not explain a deep of $> 95\%$ in its luminosity, neither the fine structures observed. Mamajek et al. (2012) discussed the possibilities of the existence of a binary star system and the eclipse having occurred due to a circumstellar disk, ruling out the following hypotheses:

- *Could a transit of circumstellar disk around a blue star in front of brighter red giant to be the origin of the eclipse?*

  There is no evidence in the spectrum of J1407 that it could contain a giant star, and such a scenario does not explain the structure observed days before and after the main deep eclipse.

- *Could a disk orbiting a compact remnant star be the cause of the eclipse?*

  J1407 is very young star (approximately 16 million years old), it would not be possible this system to contain a neutron star or a white dwarf.

- *Could the eclipse be explained by a binary star system or by J1407's own circumstellar disk?*

  The star J1407 lacks an infrared excess, which rejects the hypothesis for an accretion disk. Additionally, since the eclipse duration was approximately 56 days, the authors themselves estimated that the minimum orbital period of the secondary body should be 850 days. Thus, the ratio between the eclipse duration and the orbital period of the body would be < 0.06, significantly lower than the ratio found in binary star systems with a circumstellar disk, such as V718 Persei and KH 15 ($\sim$ 0.2 and 0.3, respectively), also rejecting this possibility.

- *Could the eclipse have occurred due to a circumstellar disk orbiting a more massive star?*

  Given the lack of infrared excess, it is difficult to consider the presence of a disk orbiting a more massive star, making it more likely that the secondary body has a lower mass than star J1407.

Ruling out all these astrophysical explanations, Mamajek et al. (2012) interpreted the presence of an extensive inclined ring system around unseen secondary companion, which it could be a brown dwarf or a giant planet matches with the observed data. The size of the ring is considerably larger than the secondary's Roche limit and occupies a significant fraction of its Hill radius. The best model of the ring has a radial extent of 0.6 au (9×10^7 Km), with a total estimated mass of 100 M_{moon}.

The figure 30 shows the light curve and model ring fit to J1407 data. Van Werkhoven et al. (2014) count at least 24 inflection points on ingress and 16 on egress, which indicate the presence of at least 24 different rings. Kenworthy and Mamajek (2015) in a more detailed analysis of the eclipse data, proposed that the system is actually composed of 37 rings, some of these inflection points were observed directly, while others were deduced through fitting the light curve on consecutive nightly observations.

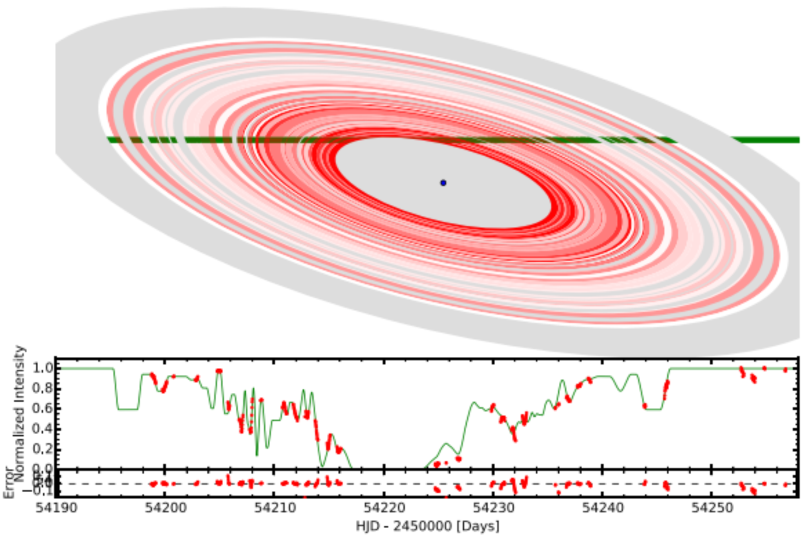


Figure 30 – The J1407b ring model fit to eclipse data. In red is J1407b's rings and the intensity of the color is proportional to the ring transmission. The path of the star J1407 behind the planet during the eclipse is illustrated in green, the thickness of this line corresponding to the star's diameter. The lower graph shows the light curve fit and red points represent the measured flux from J1407. The vertical red bars are the error bars Kenworthy and Mamajek (2015).

Kenworthy et al. (2015) explore the possibilities of a ring around a secondary body in range of [10-100] M_{Jup} and different semi-major axis to explain the dimming event. Their results shows in the case of a circular orbit, secondary body has a orbital period between 3.5 and 13.8 years, corresponding to a semi-major axis in the range of [2.2 - 5.6] au with mass < 80 M_{Jup}. Thus, this ring system is large enough to occupy a fraction > 0.15 of secondary's Hill radius and also larger than the Roche radius, confirming previous studies, which propose the existence of satellites. When consider J1407b in elliptical orbits, in cases where eccentricity > 0.7, the ring would occupy a significant fraction of the Hill radius > 0.7, making it difficult to explain the stability of the ring.

Rieder and Kenworthy (2016) investigated mass and eccentricity parameters for J1407b that matches with observed eclipse, they numerically modelled an eccentric and massive planet in range of $e = [0.6 - 0.7]$ and $m = [20 - 100]M_{Jup}$ with orbital period of 11 years. Their results strongly suggest that a retrograde ring around a secondary companion with a mass between 60 and 100 Jupiter masses.

Mentel et al. (2018) analyzed photometric data from 1890 to 1990 and from 2012 to 2018 for J1407 to search for additional eclipses, aiming to determine the orbital period of J1407b. Their analysis revealed no statistically significant eclipses. Consequently, they excluded a significant range of potential orbital periods for J1407b, specifically between 5 and 20 years. Their results show an eccentric planet with a mass range of 5 to 20 M_{Jup} could explain the eclipse if it occurred during a periapsis passage. However, the absence of further eclipses implies that J1407b might not be gravitationally bound to J1407.

Speedie and Zanazzi (2020) explored through numerical simulations the stability of a ring system around an oblate and oblique planet. Their results suggested that the likely planet's mass is > 13 M_{Jup}, assuming $J_2 = 0.5$, 30 times larger than Saturn's quadrupole coefficient, despite there no measurement of the J1407b's oblateness coefficient.

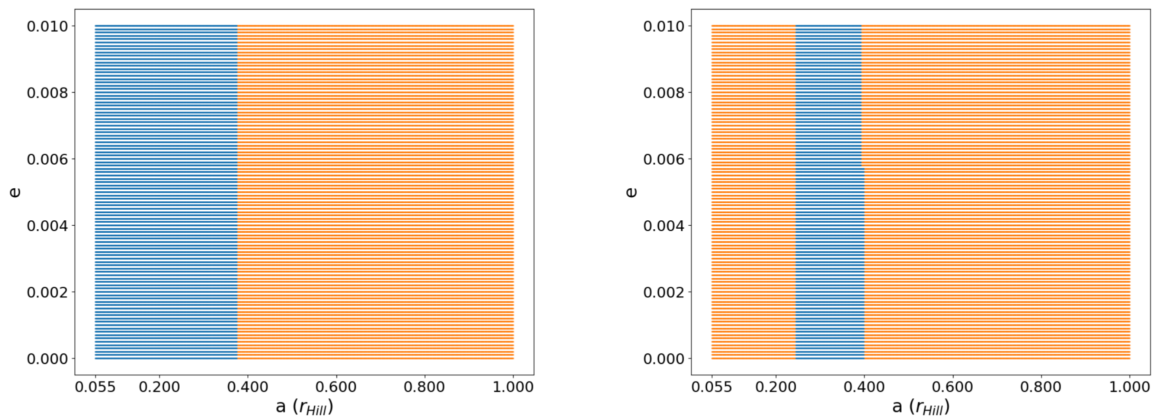
Sutton et al. (2022) conducted numerical simulations of a self-gravitating ring system around the exoplanet J1407b, assuming a mass of 20 M_{Jup} and a semi-major axis of 5 au, the simulations included rings with different eccentricities in both prograde and retrograde orientations. The prograde ring models exhibited significant perturbations within the first orbit, whereas the retrograde ring models demonstrated greater stability, especially at lower eccentricities of 0.2 and 0.4, The authors also demonstrate that an eccentric orbit is unfavorable for moon accretion, suggesting that the gap at 0.4 au does not necessarily indicate the presence of an exosatellite.

4.3 The ring model

To investigate the possibilities for observing the dimming events of stars J1407 and PDS 110 due to planetary transits and to constrain the associated physical and orbital parameters, we apply the machine learning model described in Chapter 3 to forecast stable regions around a secondary companion and estimates hypothetical eclipse duration for various ring inclinations relative to planet orbital plane (i_{ring}) and the ring plane as seen from Earth (ϕ_{ring}).

We generate various diagrams of initial conditions (a × e) for particles orbiting a planet with a mass in terms of μ ranging from 0.001 to 0.01, as mentioned in Chapter 3, 99.6% of confirmed planets have mass ratio with their star $\mu < 0.01$. The planet's eccentricity e_p ranges from 0 to 0.9, and the particle's orbital inclination $i_{ring} = [0° - 180°]$. The particle's semi-major axis (a) ranges from 1.1 planet's radius to 1 r_{Hill} with step size of $\Delta a \sim 0.0001$, while particle eccentricity e =[0 - 0.01] in steps of $\Delta e = 0.0001$, the most eccentric ring system in our solar system Uranus's ϵ ring has an eccentricity of no more than 0.01 (Melita and Papaloizou, 2020). In total, we explored 2.7×10^9 different initial conditions, which would require substantial computational effort if numerical simulations were performed.

Figure 31 shows two examples of stable maps for different hypothetical planet with $\mu = 0.05$. In the left panel, the planet has an eccentricity of 0.9 and the particle are placed in the equator plane of the planet, and in the right panel, the planet is in a circular orbit around the star, while the particles have an inclination of 60°.



(a) $e_p = 0.9$ and $i_{ring} = 0°$ (b) $e_p = 0$ and $i_{ring} = 60°$

Figure 31 – Diagram of particle initial conditions ($a \times e$) for a planetary system with $\mu = 0.05$, and different values of the planet's eccentricity and the particle's inclination. Stable and unstable particle are represented by blue and orange color, respectively.

For each diagram of initial conditions, we define a distinct ring system. The inner boundary of the stable region as the lower limit of the ring (r_{in}) and the outer boundary as the outer edge of the ring's radius (r_{out}). Assuming that the planet moves in a straight line at a constant relative velocity v_e during the eclipse, and the light curve results from a chord of the ring structure passing in front of the star, the eclipse duration t_e is given by:

$$t_e = \frac{d}{v_e} \tag{4.1}$$

where d is the size of an arbitrary chord o the ring.

In our ring model, based on Kenworthy and Mamajek (2015), we consider a ring system azimuthally symmetric, with the rings being much thinner compared to their diameter. We define ϕ_{ring} as the angle of the ring plane as seen from Earth. Here, $\phi_{ring} = 0°$ corresponds to a face-on orientation of the rings, while 90° corresponds to the rings being edge-on. The parametric equation for a projected ring, in terms of the orbital ring inclination i_{ring}, the orientation of the ring relative to our line of sight ϕ_{ring}, and the parametric variable θ, which varies from 0 to 2π is given by:

$$\begin{cases} x(\theta) = \mathrm{r}_{out}\left[\cos(\theta)\sin(\mathrm{i}_{ring}) - \sin(\theta)\cos(\phi_{ring})\cos(\mathrm{i}_{ring})\right] \\ y(\theta) = \mathrm{r}_{out}\left[\cos(\theta)\cos(\mathrm{i}_{ring}) + \sin(\theta)\cos(\phi_{ring})\sin(\mathrm{i}_{ring})\right] \end{cases} \tag{4.2}$$

where x and y are coordinates on the ring.

Given the inner (r_{in}) and outer ($\mathrm{r}_{\mathrm{out}}$) possible ring radius, the minimum angle θ needed to achieve the maximum observed chord length is d $= 2x(\theta_0)$, where θ_0 is:

$$\theta_0 = \arcsin\left(\frac{r_{in}}{r_{out}}\right) \tag{4.3}$$

The velocity of the planet can be calculated by assuming the eclipse occurs at periapsis (v_p) or apoapsis (v_a), representing the extreme orbital velocities of the planet.

$$v_p = \sqrt{\frac{1+e_p}{1-e_p}} \qquad v_a = \sqrt{\frac{1-e_p}{1+e_p}} \tag{4.4}$$

Our machine learning model is trained on the dimensionless elliptic three-body problem, where both the semi-major axis and the mean motion of the planet are normalized to 1. By inserting the values of d and v_e into equation 4.1, we can calculate the hypothetical eclipse duration in 2π units of time.

The steps of this project are summarized as follows:

- **Step 1:** Generate the stability maps ($a \times e$) for the PDS110b and J1407b systems using the trained machine learning model from Chapter 3. Vary initial conditions such as planetary mass, eccentricity, and particle parameters such as semi-major axis, eccentricity, and inclination.

- **Step 2:** Estimate the eclipse duration by assuming the size of the stable region as the ring system's extension. Use equations 4.2 and 4.1 to calculate the projected ring chord and corresponding eclipse duration.
- **Step 3:** Calculate the probability of observing the eclipse. Identify all cases matching observed eclipses for various lines of sight, considering different planetary masses, eccentricities, and ring inclinations.

4.4 Estimating eclipse duration

In this section, we estimate the probability of observing eclipses for both star PDS110 and J1407. Given specific parameters for planet mass and eccentricity, we combine different values for ring inclination ($\mathrm{i}_{\mathrm{ring}}$) and the orientation of the ring as seen from Earth (ϕ_{ring}). We then estimate the duration of the hypothetical eclipse (t_e) if the star were dimmed by a chord $d = 2x(\theta_0)$. Whenever $\mathrm{t}_e \geq$ the observed eclipse duration, we consider that this ring system could explain the dimming event.

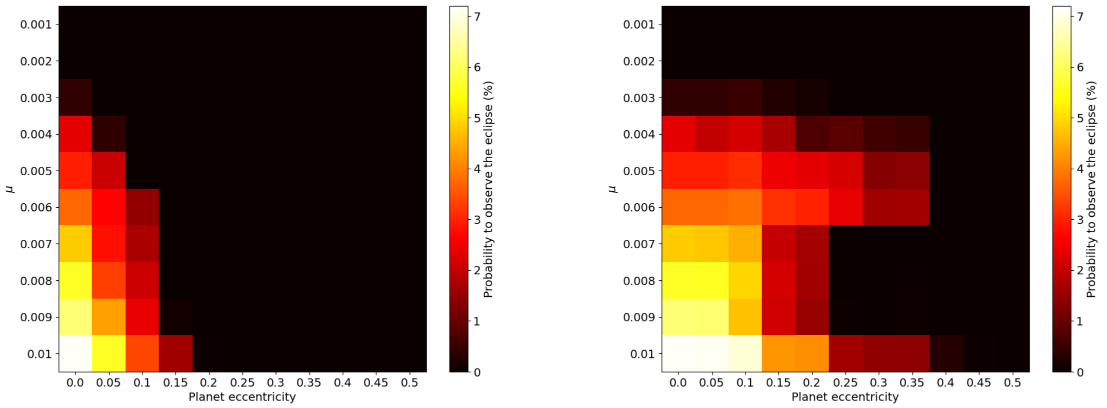
4.4.1 The PDS110b case

Pinheiro and Sfair (2021) ruled out all cases where the secondary companion eccentricity is > 0.45. Their results show that for a prograde ring the observed eclipse matches if the secondary eccentricity is < 0.05, conversely, for a retrograde ring, the eccentricity can up to 0.45. Based on these findings, we forecast stability maps for a planetary system with a mass ratio μ = [0.001 - 0.01] in step of $\Delta\mu = 0.001$, planet eccentricity $e_p = [0 - 0.5]$ with $\Delta e_p = 0.05$ and ring inclination of $\mathrm{i}_{ring} = [0° - 180°]$, with $\Delta \mathrm{i}_{ring} = 1°$.

Figure 32 shows the percentage of combinations between the angles i_{ring} and ϕ_{ring} that reproduce an eclipse lasting 25 days for a specific planet mass and eccentricity. These figures consider a planet's passage in front of the star occurring at periapsis (left panel) and apoapsis (right panel), respectively.

Our results are consistent with the findings of Pinheiro and Sfair (2021). The mass range of PDS110b is $\mathrm{m}_{\mathrm{PDS110b}} = [5 - 16.9]M_{Jup}$, corresponding to $\mu = [0.003 - 0.01]$, and the eccentricity of PDS110b is $\mathrm{e}_{\mathrm{PDS110b}} < 0.45$. As shown in figure 32, the probability of observing an eclipse with a planet having $\mu < 0.05$ (8.4 M_{Jup}) is no more than 4%. Furthermore, if the eclipse occurs at periapsis, the maximum eccentricity of the planet decreases from 0.35 to 0.05.

In the case of μ between 0.006 and 0.009 (10.1 and 15.2 M_{Jup}), the probability of observing the eclipse is between 4% and slightly more than 6% for an almost circular orbit ($e_{\mathrm{PDS110b}} < 0.1$). For planetary eccentricities > 0.1, this probability is less than 4%. Our



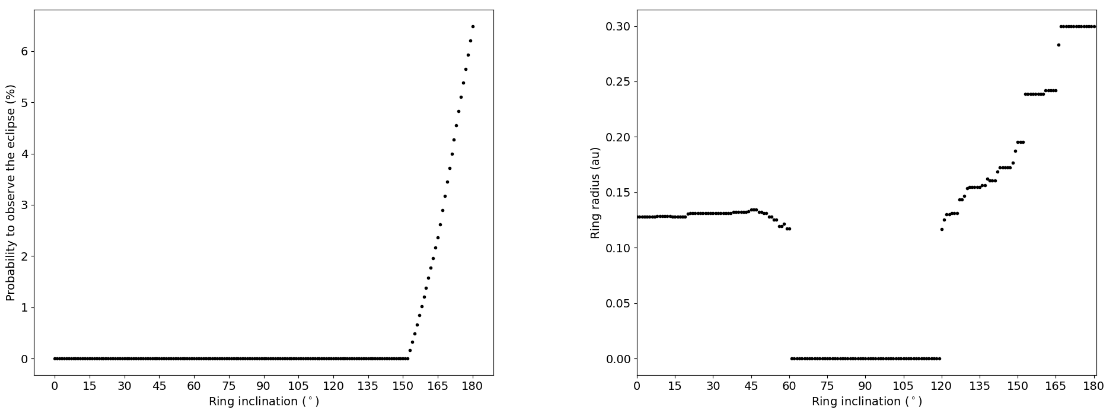
(a) periapsis

(b) apoapsis

Figure 32 – Probability to observe PDS110's eclipse for different planet configurations.

best scenario is a planet with a mass between 15.2 and 16.9 M_{Jup} in a low eccentric orbit ($e_{\mathrm{PDS110b}} < 0.1$), reaching a probability of about 7.2%.

Figure 33 focuses on our best scenario, a planet with a mass of $16.9 M_{Jup}$ in a circular orbit. The left panel shows the correlation between the probability of observing the eclipse for different ring inclinations, while in the right panel plots the ring inclination and the size of the stable region.



(a) Probability to observe the eclipse

(b) Ring radius

Figure 33 – Probability to observe PDS110's eclipse for different ring inclination and size.

Assuming the secondary companion of star PDS110 is a planet, the plot in the figure 33 shows the ring inclination must be $> 150°$, and its radius should be between 0.2 and 0.3 au. This finding is consistent with the results of Osborn et al. (2017) and Pinheiro and Sfair (2021). In any other case, the ring is not large enough to explain the eclipses happened 2008 and 2011, and for inclinations around 90°, there are no stable particles.

4.4.2 The J1407b case

The absence of a second dimming event with the star J1407 introduces uncertainty in determining the value of the secondary semi-major axis. According to Kenworthy et al. (2015), the secondary semi-major axis is likely in the range of 2.2–5.6 au, corresponding to an orbital period of 3.5–13.8 years, the authors also estimate that the largest transverse velocity is $32 \pm 2\,\mathrm{kms}^{-1}$, calculated based on the stellar radius and the steepest gradient in the light curve.

In a similar manner applied for PDS110b's ring system, we estimate the probability to observe J1407's eclipse considering the secondary body is a planet, with a range of mass ratio $\mu = [0.001 - 0.01]$ in steps of $\Delta\mu = 0.001$, an eccentricity $e_p = [0 - 0.9]$ with steps of $\Delta \mathrm{e}_p = 0.1$, and semi-major axis of $\mathrm{a}_p = [3, 5, 10, 15]$ au, for $\mathrm{a}_p > 15$ au the planet cannot achieve an 7 orbital velocity equivalent to the transverse velocity outlined in Kenworthy et al. (2015). The table 4 summarize different planet initial conditions explored in this work.

Model	**Mass (M_p)**	**eccentricity**	**semi major axis (au)**	**Orbital period (years)**
a	0.94 - 9.52	0 - 0.9	3	~ 5
b	0.94 - 9.52	0 - 0.9	5	~ 11
c	0.94 - 9.52	0 - 0.9	10	~ 31
d	0.94 - 9.52	0 - 0.9	20	~ 58

Table 4 – Orbital and physical parameters of J1407b used for the different models

Figure 34 and 35 show the probability of observing an eclipse lasting 56 days under various combinations of i_{ring} and ϕ_{ring}, where $\mathrm{i}_{ring} = [0° - 180°]$ and $\phi_{ring} = [0° - 90°]$, incremented by $\Delta i_{ring} = \Delta\phi_{ring} = 1°$. The panels in the figure 34 are scenarios with the planet's transit velocity at apoapsis, while panels of figure 35 consider velocities at periapsis. The right side of green line indicates the conditions of μ and e_p values where the planet's orbital velocity can achieve at least $30\,\mathrm{kms}^{-1}$.

A planet with $\mathrm{a}_p = 3$ au from the star reaches a maximum probability of observing an eclipse is 8.2%, assuming a circular orbit and a mass of 9.5 $\mathrm{M_{Jup}}$ ($\mu = 0.01$). For a planet with $\mu < 0.002$ and an eccentricity $\mathrm{e}_p > 0.5$, there are no configurations of i_{ring} and ϕ_{ring} that result in a 56 days transit event at apoapsis, similarly, for an eclipse at periapsis and $\mathrm{e}_p > 0.3$, a 56 days transit event is also not possible.

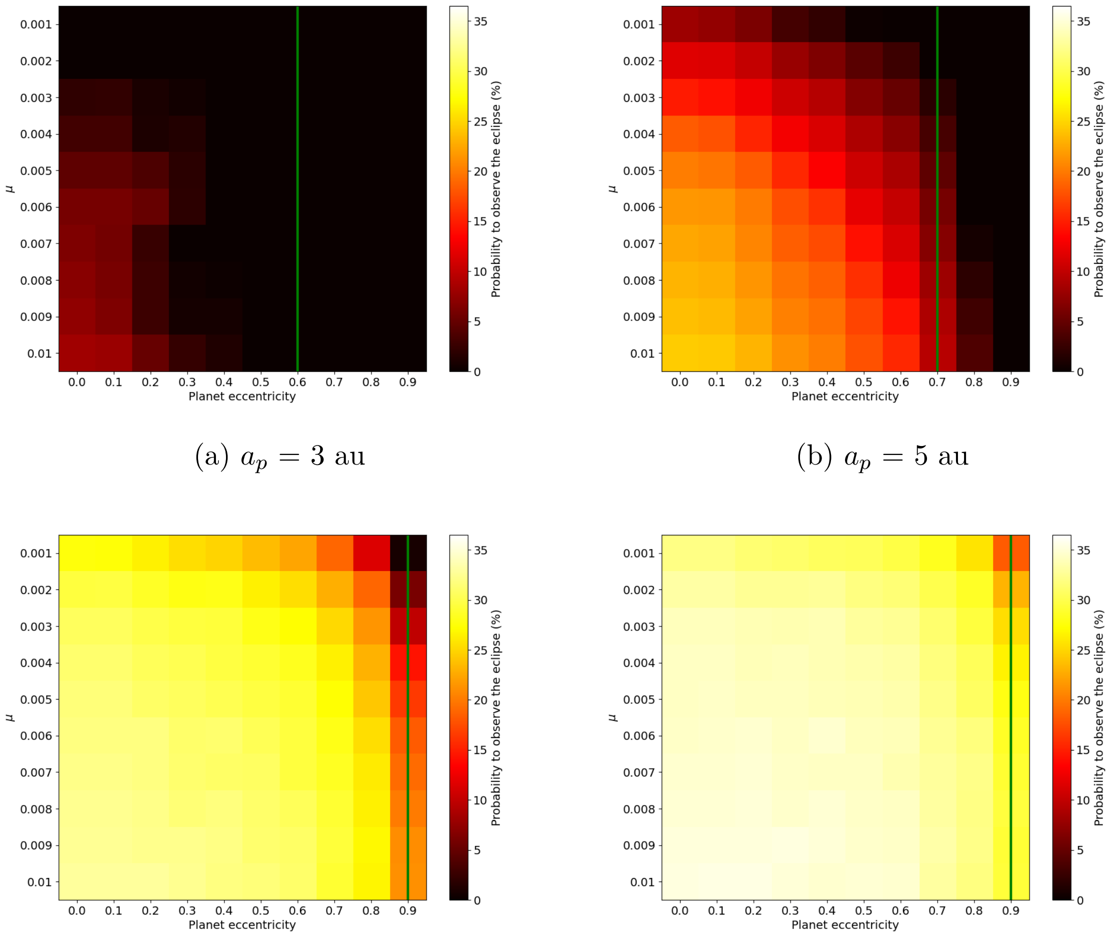
For a planet with $a_p = 5$ au from the star, a mass between 5.6 and $9.5 M_{Jup}$ ($\mu = [0.006 - 0.01]$), and an eccentricity < 0.2, the likelihood of observing J1407's eclipse can be up to three times higher compared to a planet with a semi-major axis of 3 au, ranging between 20% and 25% if the transit occurs during the apoapsis passage, and between 10% and 25% for the case of periapsis passage. However, if the planet's eccentricity

$e_p \geq 0.4$ (eclipse at periapsis) or > 0.8 (eclipse at apoapsis), the probability of observing the eclipse drops to 0%.

The green line in figure 34b demonstrate that all planets with eccentricity ≥ 0.7 can achieve an orbital velocity equivalent to transverse velocity required in Kenworthy et al. (2015), the probability of matching the eclipse can be up to $\sim 10\%$ at apoapsis, whereas this statistic drops to 0% considering an eclipse at periapsis (figure 35b).

For a planet with a semi-major axis of either 10 au or 15 au and an eccentricity $\leq$ 0.6, the probability of replicating the observed eclipse exceeds 20%, reaching a maximum of 35.4%, assuming the star being eclipsed at apoapsis, regardless of the planet's mass. Considering an eclipse occurring at periapsis, the probability up to 32.3% and drops zero when the planet eccentricity > 0.7 ($a_p = 10$ au) and > 0.8 ($a_p = 15$ au).

In both scenarios ($a_p = 10$ or 15 au), the planet's orbital velocity of $30\,\mathrm{kms}^{-1}$ is only achievable with a planet eccentricity of 0.9, in these cases, the maximum probability of reproducing the eclipse is 21.1% for $a_p = 10\,\mathrm{au}$ and 29.3% for $a_p = 15\,\mathrm{au}$.



(a) $a_p = 3$ au

(b) $a_p = 5$ au

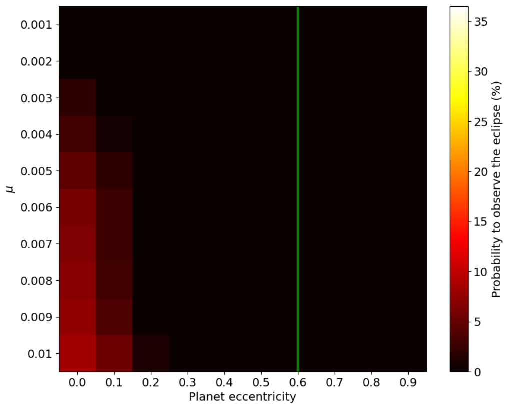
(c) $a_p = 10$ au

(d) $a_p = 15$ au

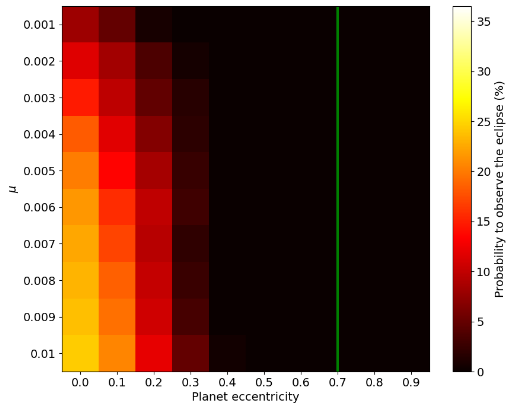
Figure 34 – Probability to observe J1407's eclipse at apoapsis for different planet configurations.

We estimate the probability of observing J1407's eclipse for different planet velocity along its orbit, assuming a planet mass of 9.52 M_{Jup}, the eccentricity (e_p) and semi-major axis (a_p) are considered as follows: $e_p = 0.7$ for $a_p = 5$ au, and $e_p = 0.9$ for $a_p = 10$ or 15 au. The figure 36 shows the variation of probabilities in function of planet velocity for the case where J1407b has a semi-major axis of 5 and 15 au.

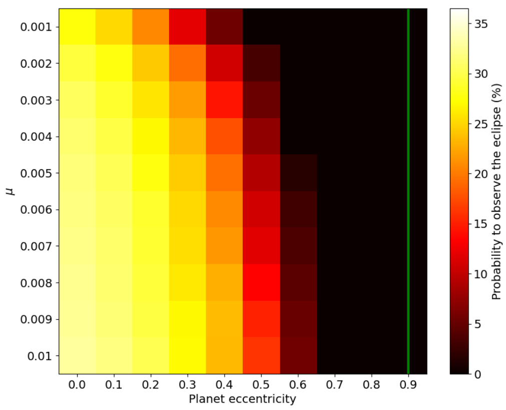
Both plots in the figure 36 show that to reproduce the eclipse observed in 2007, the planet must have a low orbital velocity, the velocity should be < 9 kms^{-1} for a planet with a semi-major axis of 5 au and < 7 kms^{-1} for $a_p = 15$ au. This suggests that the dimming event likely occurred near apoapsis. Therefore, in both cases, if the planet's velocity is faster than 30 kms^{-1}, the probability of observing an eclipse lasting 56 days is zero, indicating that the mass ratio of this system is $\mu > 0.01$, suggesting that the secondary companion of J1407 is likely a brown dwarf.



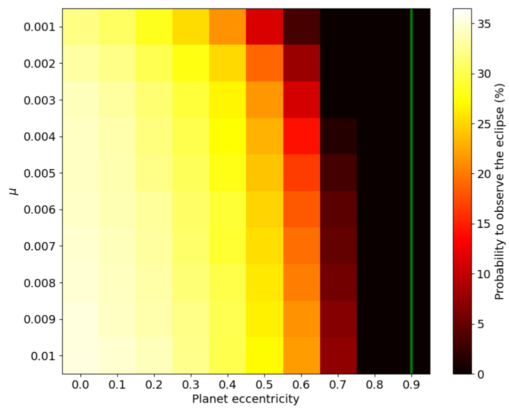
(a) $a_p = 3$ au



(b) $a_p = 5$ au

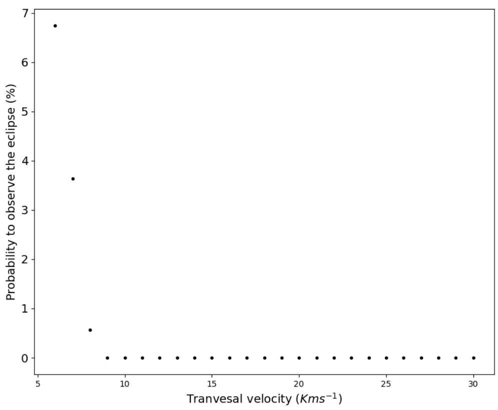


(c) $a_p = 10$ au

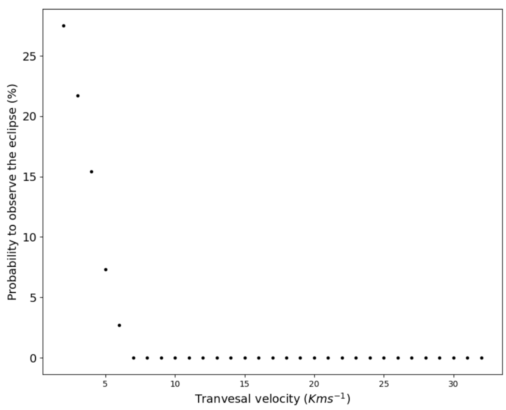


(d) $a_p = 15$ au

Figure 35 – Probability to observe J1407's eclipse at periapsis for different planet configurations.



(a) $a_p = 5$ au, $e_p = 0.7$ and 9.5 M_{Jup}



(b) $a_p = 15$ au, $e_p = 0.9$ and 9.5 M_{Jup}

Figure 36 – Probability to observe J1407's eclipse for different transversal velocity.

4.5 Final remarks

The first exoring candidate was reported by Mamajek et al. (2012), who observed a dimming event with the star J1407 in 2007, this event lasted approximately 56 days and caused a dimming of over 95%. The phenomenon may be attributed to a giant ring system extending up to 0.6 au around a secondary companion, which could be a brown dwarf or a giant planet. There is a gap at 0.4 au within the ring system, suggesting the presence of an exosatellite (Kenworthy and Mamajek, 2015).

Another exoring candidate is around PDS110b, Osborn et al. (2017) detected two similar eclipses lasting 25 days with star PDS110 in 2008 and 2011, indicating a possible circum-secondary disc structure that extend up a radius of 0.3 au around a object with mass up to 70 M_{Jup} and semi-major axis of $\sim$ 2 au.

This study aims to constrain the planet mass and eccentricity, as well the size, and inclination of the rings associated with the eclipsing secondary discs around J1407b and PDS110b, assuming the star-planet system has $\mu \leq 0.01$. A machine learning model was employed to predict stable particles varying different initial orbital and physical conditions of these systems. Overall, we explored 2.7×10^9 different initial conditions, which require a lot of computational effort if the machine learning model were not applied. The stability maps enable us to determine the maximum possible size of the ring and calculate hypothetical eclipse duration, as well to estimate the probability to reproduce the observed events.

Analyzing the planetary ring system around PDS110b, our findings show a star-planet system with $\mu > 0.04$ and planet with eccentricity < 0.3 can host a ring system that reproduce the observable eclipse. However, the most probable case is a planet with a mass between 15.2 and 16.9 M_{Jup} in a circular or almost circular orbit ($e_p < 0.1$) show the highest probabilities to match with the observed eclipses in 2008 and 2011 reaching around 7.2%. Additionally, the ring inclination likely is $> 150°$, and possible ring's radius ranging between 0.2 and 0.3 au, consistently with Osborn et al. (2017) and Pinheiro and Sfair (2021)'s results.

The dimming event observed in the star J1407 can be explained by a transit of a planet with a semi-major axis between 3 and 15 au, with probabilities reaching up to 36%. For a planet with a semi-major axis > 5 au and high eccentricity ($e_p > 0.7$), the orbital velocity can reach the transversal velocity of 32 ± 2 kms^{-1} estimated by (Kenworthy et al., 2015). Observational estimations suggest that eclipses cannot be observed for a planet with a transverse velocity > 9 kms^{-1} with planet mass ≤ 9.52 M_{Jup}.

5 The resilience of the sailboat stable region

Before the arrival of the New Horizons probe in Pluto-Charon system, Giuliatti Winter et al. (2013) and Giuliatti Winter et al. (2014) conducted several numerical simulations to identify stable regions present in this binary system, which has a mass ratio of $\mu = 0.1165$ and a separation distance of d = 19,571 km. Their findings showed a peculiar stable region orbiting around Pluto known as the sailboat region, located midway between the massive bodies and have high eccentricities. The sailboat region is confined within the interval of initial particle orbital semi-major axis (a) and eccentricity (e) of a = [0.5d - 0.7d] and e = [0.2 - 0.9], respectively.

This region received this name, because it resembling a sailboat shape in the initial conditions diagram (a $\times$ e), see figure 37. Giuliatti Winter et al. (2014) studied in detail the sailboat region for Pluto-Charon binary system. They numerically simulated a ensemble of test particles perturbed gravitationally by Charon, assuming that Pluto's other satellites Nix, Hydra, Kerberos, and Styx had negligible effects on the particle's dynamics. They used the Poincaré surface of section to confirm the stability of particle trajectories within the sailboat region and identified it is associated with family BD of periodic orbits described by Broucke (1968).

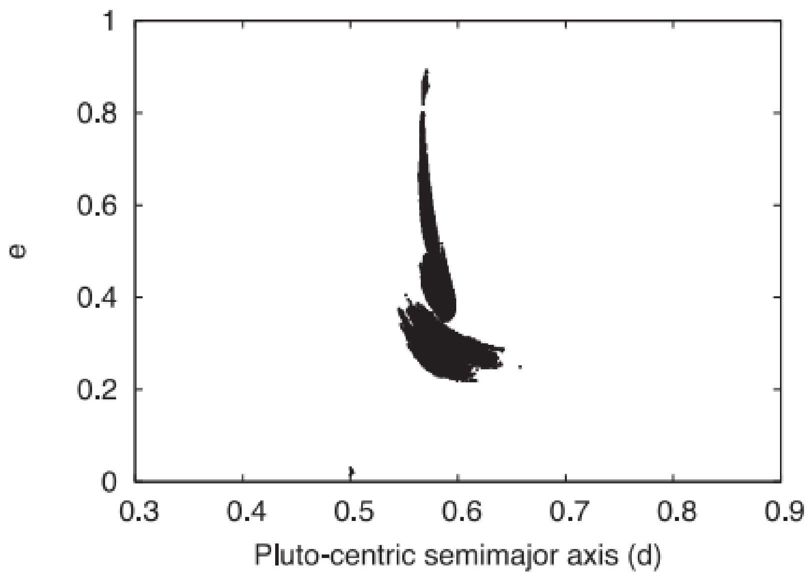


Figure 37 – Diagram of the initial conditions (a $\times$ e), showing the sailboat region in the Pluto-Charon system. All the others particle orbital element were set to be zero (Giuliatti Winter et al., 2014).

Giuliatti Winter et al. (2014) analyzed the extension of the sailboat region for different values of particle orbital inclination (i) and argument of pericenter (ω). Their results showed that the sailboat region survives within two narrow intervals of ω: the first ranging from ω = -10° to 10°, and the second from ω = 160° to 200°, being its maximum size occurs at ω = 0° and 180°. The study on inclination indicated that the sailboat region exists for i $\leq$ 90°. Giuliatti Winter et al. (2013) demonstrated that the sailboat region is the most resilient stable region in the system concerning inclination.

These findings were combined in another paper by Winter et al. (2015) to provide insights for determining safe trajectories for the New Horizons mission. The probe passed near this region on July 14, 2015.

This work aims to deepen our understanding of the stability of the sailboat region by varying orbital and physical parameters, thereby constraining the range where the sailboat can exist in binary system. These parameters include the binary mass ratio, the eccentricity of secondary body, particle orbital inclination, and the argument of pericenter.

We conducted thousands of numerical simulations of three-body problems, exploring several initial conditions. These simulations were used to create a dataset for training a machine learning algorithm that predicts no simulated stable particles around the primary body. This approach significantly reduces computational efforts.

This chapter is organized as follows: the section 5.1 discusses different types of binary systems, including binary star and dwarf planets. The section 5.2 describes the numerical method implemented. In the sections 5.3, 5.4, 5.5 and 5.6 present our machine learning model predictions for each different parameters analyzed: such as mass ratio of the system, secondary body eccentricity, particle inclination, and argument of pericenter, respectively. In final section 5.7 contains our final remarks.

5.1 Binary Systems

The majority of observed stars are predominantly in binary systems or belongs to a systems with three or more stars. Binary stars consist of two stars orbiting a shared center of mass that are outside of them. The nature of the interaction between the stars determines whether they are classified as detached, semi-detached, or contact binaries.

In detached binaries, both stars are within their Roche lobes, which is the region around each star where its gravitational forces dominate, allowing them to orbit without sharing material. Semi-detached binaries, one of the star filled its Roche lobe, enabling it to transfer material to its companion star. The last type is contact binaries that occur when both stars fill their Roche lobes, leading mass exchange between them (Kopal, 1955).

A survey with 489 eclipsing binary star systems, utilizing data from three catalogs, categorizes the systems into 100 contact star type from Catalog of Early-type Contact Binaries (Bondarenko and Perevozkina, 2004), 232 semi-detached types from Catalog

Semi-detached Eclipsing Binaries (Surkova and Svechnikov, 2004), and 157 detached types from Catalog of DMS-type Eclipsing Binaries, (Svechnikov and Perevozkina, 2004).

Figure 38 shows the semi-major axis of secondary star (a_s) expressed in units of the primary star's radius (r_p) versus the binary mass ratio (μ). Most binary stars with a mass ratio close to that of the Pluto-Charon system (the black point) belong to a semi-detached type. Due to mass transfer between the stars in these systems, the region between these stars is likely unstable. However, there are four detached systems with a mass ratio < 0.25 and separation distances ranging from 2 $r_p$to 8 r_p, which likely contain a sailboat region. Giuliatti Winter et al. (2014) showed that the sailboat region can exist even for different binary separations.

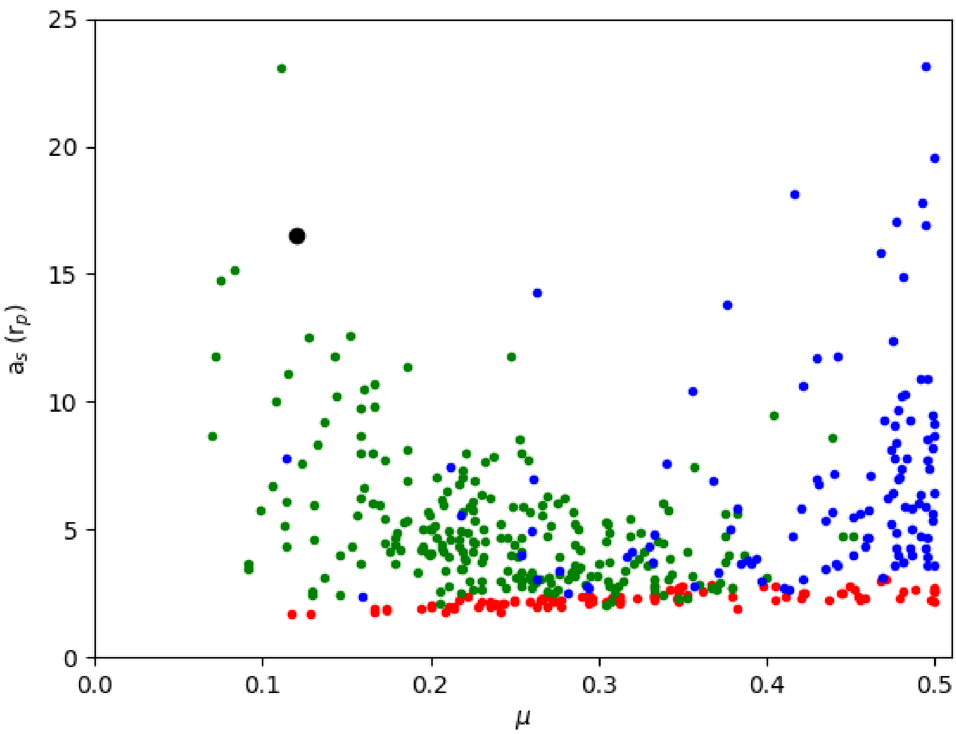


Figure 38 – Diagram of a binary star system ($a_s \times \mu$). The diagram categorizes the binaries type as followed: Detached stars in blue, Semi-Detached type in green, and Contact type in red color. The black point represents the Pluto-Charon system.

5.1.1 Dwarf Planets in our Solar System

The term "dwarf planet"was introduced during the 26th General Assembly of the International Astronomical Union (IAU). Unlike planets, objects in this new category have not cleared the surrounding region of their orbits. Table 5 lists the ten largest confirmed or candidate dwarf planets that host at least one satellite. This table includes information such as the mass ratio of the system, the satellite's orbital semi-major axis (a), and its eccentricity (e).

Binary System	μ	**a (km)**	**e**
Pluto and Charon[a]	0.12	19600	0.000161
Eris and Dysnomia[b,c]	0.0050	37273	0.0062
Haumea and Hi'iaka[d]	0.012	49880	0.0513
Makemake and S/2015 (136472) 1[d]	0.002	21100	unknown
Quaoar and Weywot[e]	0.00433	13289	0.056
Gonggong and Xiangliu[f]	unknown	24021	0.2908
Orcus and Vanth[b]	0.16	8999.8	0.00091
Salacia and Actaea[g]	0.033	5724	0.0098
Varda and Ilmarë[h]	0.083	4809	0.0181
(532037) 2013 FY27 system[i]	unknown	9800	unknown

Table 5 – Binary system of dwarf planet and their satellites in our Solar System. Font: (a) Brozović and Jacobson (2024), (b) Brown and Butler (2023), (c) Holler et al. (2021), (d) Ragozzine and Brown (2009), (e) Morgado et al. (2023), (f) Kiss et al. (2019), (g) Grundy et al. (2019), (h) Grundy et al. (2015), (i) Sheppard et al. (2018).

All the listed bodies are Trans-Neptunian objects (TNOs). Among the confirmed dwarf planets, only Ceres and Sedna have no discovered satellites, while Pluto contains five satellites, and Haumea has two: Hi'iaka and Namaka. Additionally, Quaoar and Haumea host ring systems, both ring system close to the 1/3 spin–orbit resonance with the primary body and in the case of Quaoar its two rings are located outside its Roche limit (Pereira et al., 2023; Morgado et al., 2023; Ortiz et al., 2017).

Among all objects presented in table 5, Pluto and Charon are unique in having been visited by a spacecraft. The Orcus-Vanth and Varda-Ilmarë systems have mass ratios similar to that of Pluto and Charon, with $\mu = 0.16$ and 0.083, respectively. This suggests that both binaries likely exhibit a sailboat region similar to the Pluto system.

Orcus is a Plutino and a Kuiper Belt Object (KBO) locked at 2:3 orbital resonance with Neptune. The Orcus-Vanth system is unusual for the Kuiper Belt, unlike most large KBOs which have small satellites, Vanth is relatively large compared to Orcus (Sheppard, 2012). Orcus has a diameter of 910 km, while Vanth's diameter is 475 km. This system mass is 6.32×10^{20} kg and their densities are 1.4 g/cm^3 for Orcus and 1.5 g/cm^3 for Vanth (Brown et al., 2010).

Varda is an object with a diameter of 705 km and a density of 1.27 g/cm^3. Ilmarë, meanwhile, has approximately half the size of Varda with 361 km. According to Grundy et al. (2015) both have the same density and the orbital period of Ilmarë is 5.75 days, similar with period of Charon around Pluto (approximately 6.5 days).

Investigating the stability regions around these bodies can be helpful for future space missions and observation campaigns, as it may identify the most probable locations to discover new objects or dust ring systems.

5.2 Numerical Simulation

To investigate the resilience of the sailboat region for different binary systems, we numerically simulated several elliptic three-body problem, exploring a range of initial orbital conditions and the binary mass ratio. Each simulation involved a dimensionless binary system, with a test particle orbiting the primary body in an S-type orbit and gravitationally disturbed by secondary body.

In all numerical simulations, the primary body was modeled as a point mass with a mass M_p= 1 - μ, where μ is the mass ratio of the system. The secondary body had a mass m_s= μ, a semi-major axis a_s= 1, and a collision radius (r_s) defined as 10% of its Hill radius. The Hill radius is given by:

$$r_s = 0.1 \left[(1 - e_s) \sqrt[3]{\frac{\mu}{3}} \right] \tag{5.1}$$

where $e_s$is the eccentricity of the secondary body.

We numerically simulated 300,000 hypothetical binary systems, randomly selecting the mass ratio (μ), test particle semi-major axis (a), and eccentricity (e) from uniform distributions within the following ranges: μ = [0.01 - 0.3], a = [0.45 - 0.7], and e = [0 - 0.99], respectively. All other orbital elements for both the secondary body and the particle were set to zero.

The numerical results were used to create our first dataset, which trained a machine learning algorithm to classify whether a particle with given initial conditions is stable or unstable. The goal was to identify the critical value of the binary mass ratio where the sailboat region exists.

We expanded our analysis to examine how the sailboat region's extent varies with different initial eccentricities of the secondary body, particle orbital inclination, and argument of the pericenter. We generated three new datasets, each comprising 300,000 samples, varying the mass ratio of the system, particle semi-major axis, eccentricity, and an additional parameter:

1. One dataset included variations in the eccentricity of the secondary body (e_s), ranging from 0 to 0.99.

2. Another dataset involved changes in the particle's orbital inclination, spanning i = $[0° - 180°]$.

3. A third dataset examined variations in the particle's argument of pericenter, covering a range of $\omega = [0° - 360°]$.

We numerically integrated using the REBOUND package with the IAS15 integrator (Rein and Spiegel, 2014) over a timespan of $10^4 \times 2\pi$ units of time.

Whenever the particle collided with the secondary body or was ejected from the system, the numerical simulations were stopped. Ejection occurred when a particle achieved a hyperbolic orbit (e $\geq$ 1) or when its semi-major axis a $>$ 1, particles that survived throughout time of simulation were considered stable.

All the four dataset were used to train a machine learning algorithm XGBoost (Extreme Gradient Boosting), in chapter 3 XGBoost demonstrated superior performance in solving the three-body problem. Each dataset underwent cross-validation techniques to assess algorithm performance with and without resampling methods presented in the chapter 2, random undersampling and oversamplinbg, SMOTE, Boderline-SMOTE and ADASYN. Hyperparameter tuning and threshold adjustments were systematically applied to optimize model performance specific to each dataset, resulting in the identification of the best models for the given data scenarios.

After training with these four datasets and finding the best machine learning model for each of them, we forecast 3,627,855,000 different initial conditions, making it unfeasible to achieve the same results through numerical simulations.

5.3 Analyse of Sailboat region for different binary mass ratios

In this section, we to verify how the sailboat region changes in relation to the mass ratio of the system. The first dataset analyzed was composed by three freedom parameters: binary mass ratio μ, particle semi-major axis (a) and eccentricity and (e).

The overall outcome of this dataset has 300,000 samples, being 77.98% of them computed collision, 0.75% the particle were ejected from the system, and 21.27% the particle remained stable throughout the simulations. To tackle with this imbalanced class distribution, random oversampling showed most effective among the implemented resampling methods.

Our best model achieved an accuracy of 98.54%, However, in imbalanced classification scenarios, accuracy alone may not adequately reflect the model's performance for minority classes. Instead, metrics like recall and precision offer better insights.

For the stable class, our model achieved a recall of 97%, while for the unstable class, the recall was 99%, indicating that it correctly identified 97% of all stable samples and 99% of unstable particles.

Our model achieved a precision of 97% for stable class, and 99% for unstable class, meaning that 97% of samples classified as stable and 99% of samples classified as unstable were correctly predicted.

The machine learning forecast results are shown in the Figures 39 ans 40, which present the diagrams of initial conditions (a $\times$ e) for a test particle orbiting the primary in binary systems with various mass ratios in the range of μ = [0.01 - 0.24], with a step size of $\Delta\mu = 0.01$. Furthermore, we have generated stability maps for the case where μ = [0.25 - 0.30]. Each diagram there are 247,500 initial conditions, where the particle semi-major axis and eccentricity vary in the interval of a = $[0.45 - 0.7]$ and e = $[0 - 0.99]$, both with step size of Δa = Δe = 0.001.

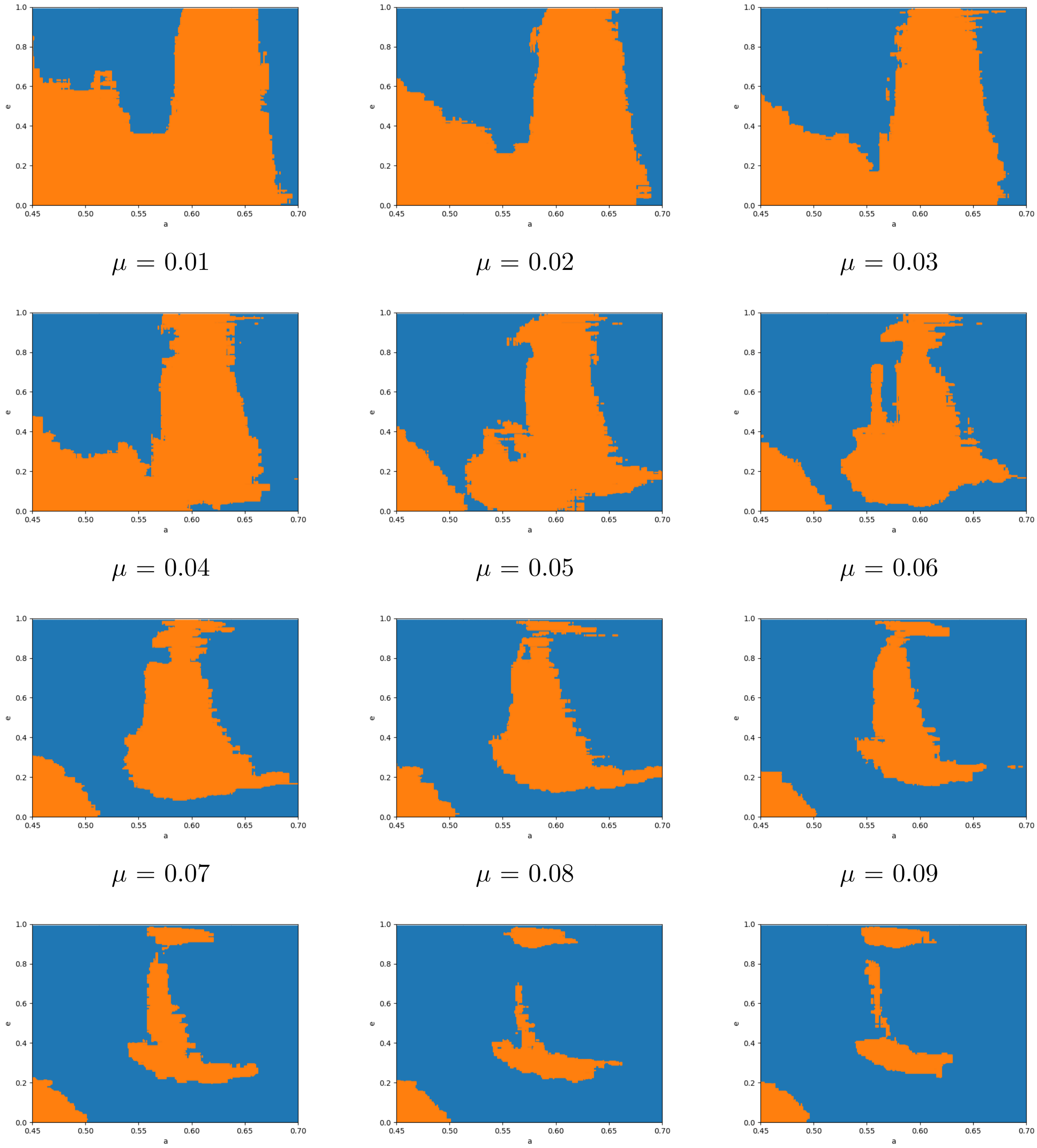
Binary systems with $\mu \leq 0.04$ have a stable region orbiting the primary body that extends up to approximately 0.7. Within the interval a = [0.55 - 0.68], the stable particles achieve high eccentricities, slightly less than 1.

When the binary mass ratio reaches $\mu = 0.05$, the stable region splits into two distinct regiaons. The first region is defined by a < 0.52 and e ≤ 0.4, while the second one is in a range of a = [0.52 - 0.7] and e < 1, known as the sailboat region. As the μ value increases, the sailboat region gradually diminishes in size until it completely disappears, reaching a maximum mass ratio of $\mu = 0.22$.

In the sailboat region, the minimum eccentricity of stable particles increases with higher μ values. For a binary system with $\mu = 0.07$, the sailboat region has a minimum eccentricity of e ~ 0.1, when $\mu = 0.09$, the minimum eccentricity rises to about e ~ 0.18. At $\mu = 0.10$, the sailboat region splits into two regions, the larger one is in the range a = $[0.52 - 0.67]$ and e = [0.2 - 0.8], while the smaller region consists of very eccentric orbits with e > 0.85.

For binary systems with $\mu = 0.15$, the main region of the sailboat is confined to a = [0.52 - 0.63] and e = [0.22 - 0.7], this region becomes less eccentric and loses its sail-like shape. When μ increases to 0.16, the maximum eccentricity decreases to about e $\sim$ 0.6, for $\mu = 0.18$, the stable region is found within a = [0.5 - 0.6] with e $<$ 0.5, indicating a gradual shift closer to the primary body and a reduction in maximum eccentricity.

At $\mu = 0.22$, new stable regions appear at a = [0.5 - 0.6] with e < 0.5, this region expands and merges with the remaining sailboat region at $\mu = 0.23$, where the semi-major axis ranges from 0.5 to 0.52 and e ~ 0.4.



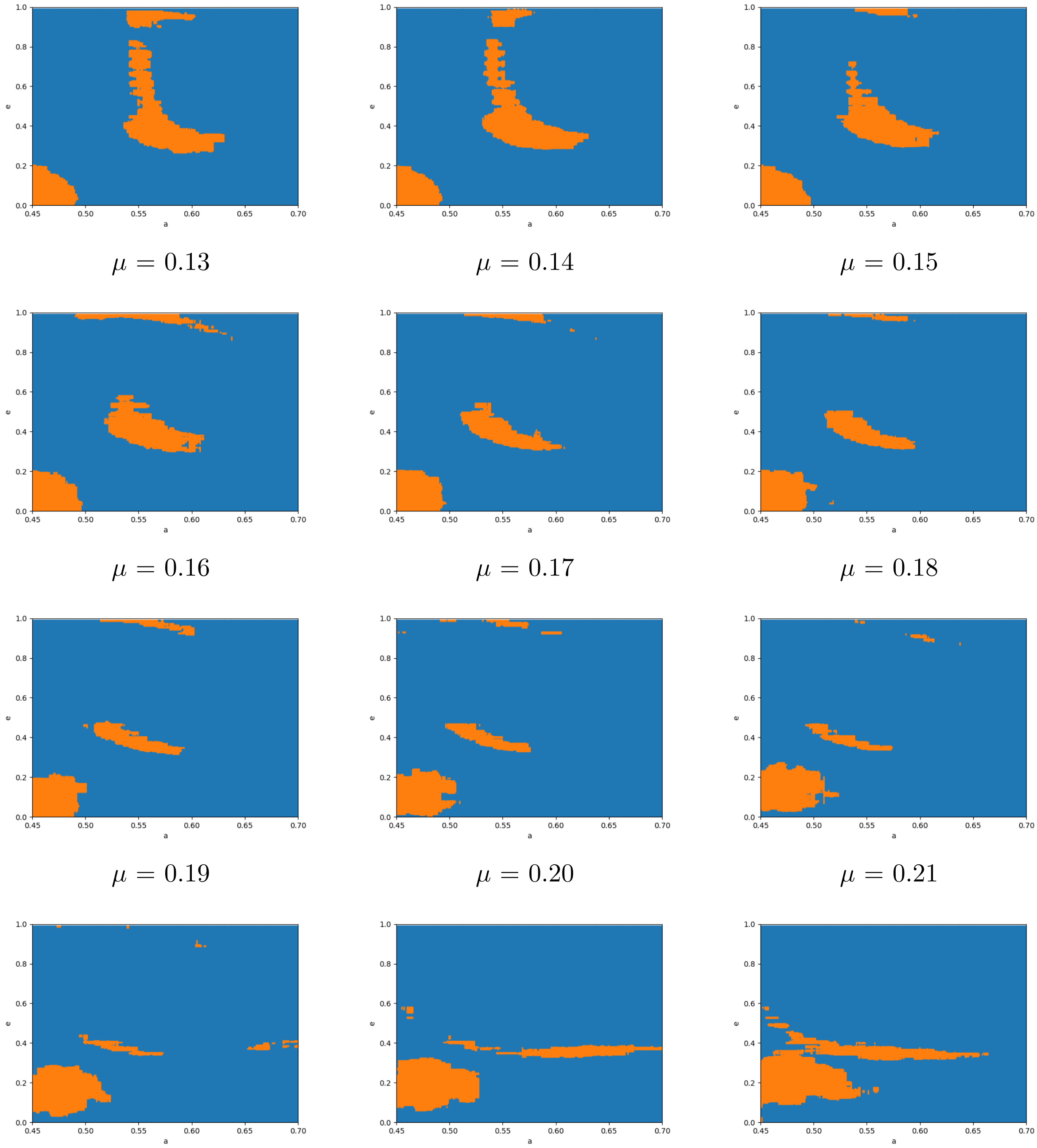
$\mu = 0.01$ $\mu = 0.02$ $\mu = 0.03$

$\mu = 0.04$ $\mu = 0.05$ $\mu = 0.06$

$\mu = 0.07$ $\mu = 0.08$ $\mu = 0.09$

$\mu = 0.10$ $\mu = 0.11$ $\mu = 0.12$

Figure 39 – Stability regions (in black) for different mass ratio values μ = [0.01-0.12] and particles in coplanar orbit around primary body, with others orbital elements $\omega = \Omega = \mathrm{f} = 0°$.



$\mu = 0.13$ $\mu = 0.14$ $\mu = 0.15$

$\mu = 0.16$ $\mu = 0.17$ $\mu = 0.18$

$\mu = 0.19$ $\mu = 0.20$ $\mu = 0.21$

$\mu = 0.22$ $\mu = 0.23$ $\mu = 0.24$

Figure 40 – Stability regions (in black) for different mass ratio values μ = [0.13-0.24] and particles in coplanar orbit around primary body, with others orbital elements $\omega = \Omega = \mathrm{f} = 0°$.

Some eccentric stable orbits (e > 0.85) in the sailboat region can become unstable depending on the size of the primary body. Figure 41 illustrates the sailboat region for a binary system with $\mu = 0.12$ (the closet μ with the Pluto-Charon system), the red dashed line indicates where the pericenter of the particle matches the radius of Pluto. As shown in the figure, all particles with e > 0.9 will collide with Pluto and become unstable. This finding aligns with the numerical results of Giuliatti Winter et al. (2014), who constrained the sailboat region within a = [0.5 - 0.7] and e = [0.2 - 0.9].

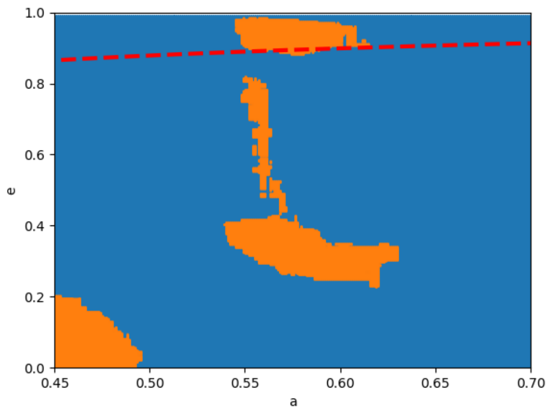


Figure 41 – The sailboat region for a binary system with $\mu = 0.12$ (the closest case with Pluto-Charon system, where Charon has eccentricity). The red dashed line represents the pericenter of the particle in the same distance of Pluto's radius. There is other stable regions in the range $a = [0.3 - 0.7]$, not showed in the plot.

The stability of the sailboat region has been confirmed through Poincaré surface of section analysis conducted by Dr. Ernesto Vieira, a coauthor of this project. Although these results are not included in this text, they will be featured in a paper currently in the final stages of writing, "The Resilience of the Sailboat Stable Region".

5.4 Eccentric secondary body case

In this section, we investigate the stability of the sailboat region by varying the eccentricity of the secondary body (e_s). The dataset used in this stage includes four freedom parameters:

- Mass ratio of the system: μ = [0.01 - 0.23]
- Particle semi-major axis: a = [0.45 - 0.7]
- Particle eccentricity: e = [0 - 0.99]
- secondary body eccentricity: e_s= [0 - 0.99]

By exploring these parameters, we aim to understand how changes in the secondary body's eccentricity affect the stability of the sailboat region. The overall outcome from the dataset is as follows: 1.77% of the samples are stable, while 98.23% are classified as unstable. Among the unstable particles, 95.39% were ejected, and 2.84% collided with a secondary body.

To address the class imbalance, we applied ADASYN (Adaptive Synthetic Sampling) oversampling technique, which yielded the best performance in our best machine learning model. As discussed in chapter 2, ADASYN generates synthetic samples for the minority class based on the sample distribution in the feature space, thus enhancing the representation of the minority class in the training dataset (He et al., 2008).

Our best machine learning model achieved an accuracy of 99.55%. For the stable class, the recall was 87%, while for the unstable class, the precision was nearly 100%. In a test dataset consisting of 60,000 samples, the model correctly identified 913 as stable and 58,817 as unstable, with only 270 misclassifications. Specifically, 135 stable particles were misclassified as unstable, and 135 unstable particles were misclassified as stable.

We employed machine learning to forecast stability maps (a $\times$ e) with $\Delta\mu = 0.01$ and $\Delta e_s = 0.001$, defining the maximum value of e_s at which the sailboat region completely vanishes. Figure 42 shows the maximum eccentricity of the secondary body (blue dots) for various mass ratios where the sailboat region exists. The red curve represents a fit obtained using the Least Squares method, each black square corresponds to a binary system of dwarf planets and their satellites, as listed in Table 5, while the green square represents the Pluto-Charon system.

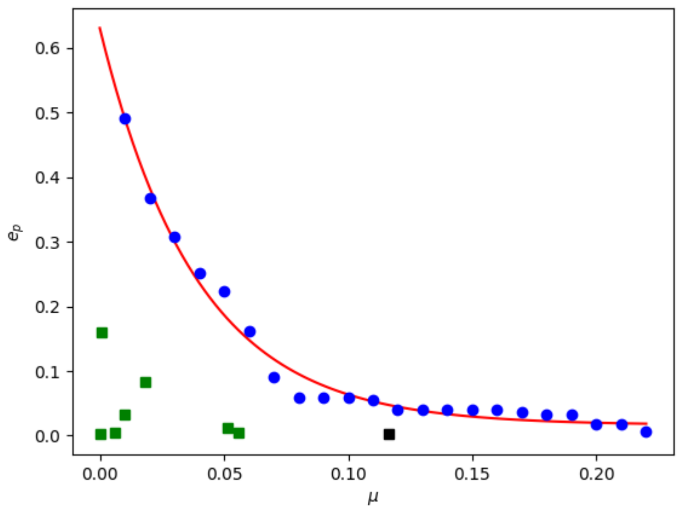


Figure 42 – The maximum eccentricity of secondary body (blue dots) versus mass ratio of the system. The red curve represents a fitted model obtained using the Least Squares method. Black squares denote binary systems of dwarf planets and their satellites, while the green square represents the Pluto-Charon system.

Even small changes in the eccentricity of the secondary body are sufficient to substantially reduce the extent of the sailboat region. The maximum eccentricity ($e_{\max}$) value can be empirically expressed as

$$e_{\max} \approx (0.016 \pm 0.006) + (0.614 \pm 0.023)e^{(-25.626 \pm 1.636)\mu} \tag{5.2}$$

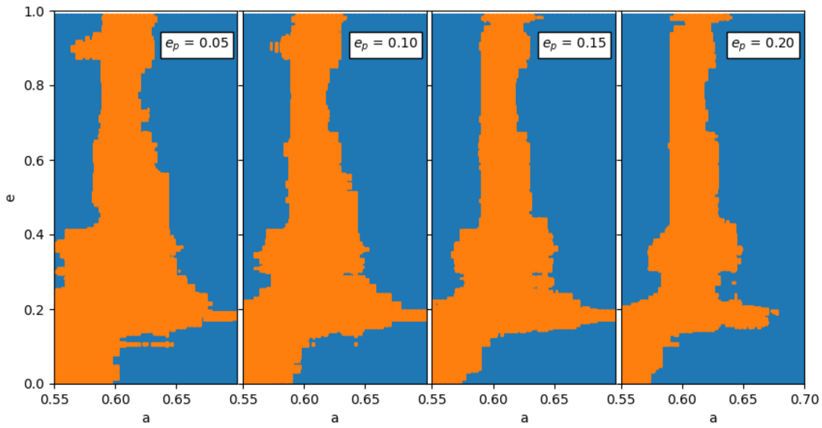
with correlation coefficient of $\chi^2 = 0.983$.

This equation indicates a significant exponential decrease between the maximum eccentricity (e_{max}) and the parameter μ. Figure 42 shows that all the dwarf planets listed in Table 5 may exhibit a sailboat region, suggesting potential regions for hosting satellites or dust ring systems. It is even possible to design mission with a spacecraft studying the system from an orbit of this region, due to its stability.

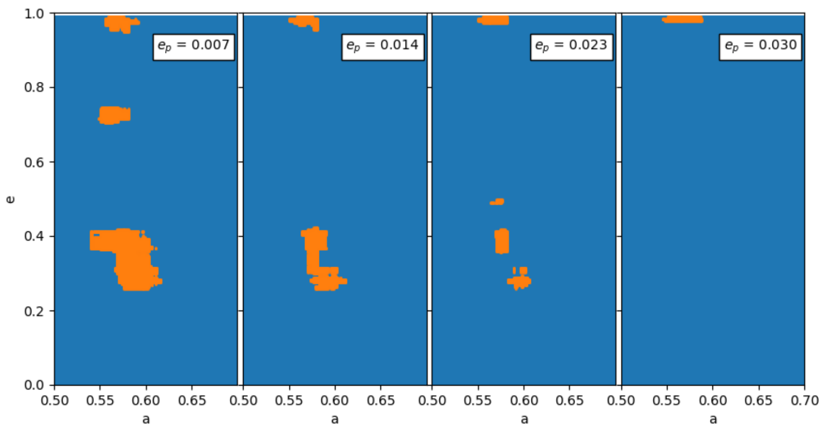
Figure 43 shows the evolution of the sailboat region as the secondary body eccentricity changes for binary systems with mass ratios of $\mu = 0.05$, 0.12 and 0.22.

As the eccentricity of the secondary body increases, particles with lower eccentricity within the sailboat region become unstable. In Figure 43, for $\mu = 0.05$ and $e_s = 0.2$, only particles with eccentricities around e ~ 0.8 remain stable. In the case of binary of $\mu = 0.12$ with $e_s = 0.03$ and $\mu = 0.22$ with $e_s = 0.007$, only particles with eccentricities close to 1 survive, although these particles are likely to eventually collide with the primary body.

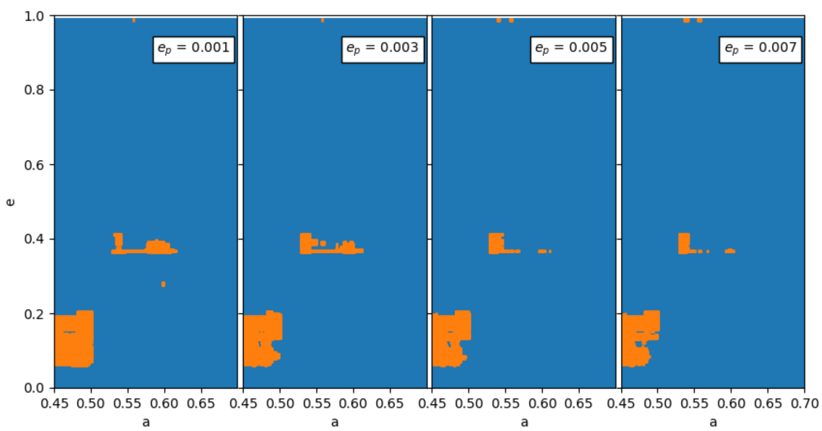
Considering that particles with e > 0.9 become unstable due to collisions with the primary body, the maximum allowable eccentricity of the secondary body exhibits significant changes for certain μ values compared o the results shown in figure 42. For μ values ranging from 0.12 to 0.16, the maximum eccentricity e_{max} changes from 0.04 to around $e_{max} =$ [19 - 0.25]. When $\mu = 0.17$ the $e_{max} = 0.036$, although excluding the case where particle eccentricity > 0.9 the $e_{max} = 0.019$, showing a significant difference. In the case $\mu = 0.18$ and $\mu = 0.19$, the e_{max} values drop to 0.014, for a binary with $\mu = 0.20$ and $\mu = 0.21$, the $e_{max} = 0.014$. Lastly, for $\mu = 0.22$, the e_{max} is 0.007.



(a) $\mu = 0.05$



(b) $\mu = 0.12$



(c) $\mu = 0.22$

Figure 43 – Stability maps ($a \times e$) showing the evolution of sailboat region with changing secondary body eccentricity, for binary systems with mass ratios of $\mu = 0.05$, 0.12 and 0.22.

5.5 Orbital particle inclination case

Aiming to investigate the extent of the sailboat region under different initial orbital inclinations of particles (i), we constructed a dataset comprising 300,000 samples. This dataset includes parameters such as the μ and the particle's orbital elements a, e, i, which were varied as follows: : μ = [0.01 - 0.23], e = [0 - 0.99], a = [0.45 - 0.7], and i = [0° - 180°].

The numerical outcome are 20.11% of the systems in this dataset are stable, while 79.89% are unstable. Among the unstable systems, 79.75% of particles were ejected, while only 0.14% recorded a collision. Our best machine learning model employs the Synthetic Minority Oversampling Technique (SMOTE). This method creates synthetic samples for the minority class within the training data based on the Euclidean distance of nearest neighbors, using a factor between 0 and 1 (Chawla et al., 2002).

Our best model performance achieved an accuracy of 97.15%, with a precision of 93% for the stable class and a recall of 98% for the unstable class. Utilizing this model, we generated diagrams of initial conditions (a $\times$ e) for particle inclination ranging from i = 0° to 180°, incrementing by Δi = 1°. This process was repeated for each $\mu \leq 0.22$, with a step size of $\Delta\mu = 0.01$. The figure 44 shows where stable region can be found within the range a = [0.45 - 0.7] for different values of μ and particle orbital inclination.

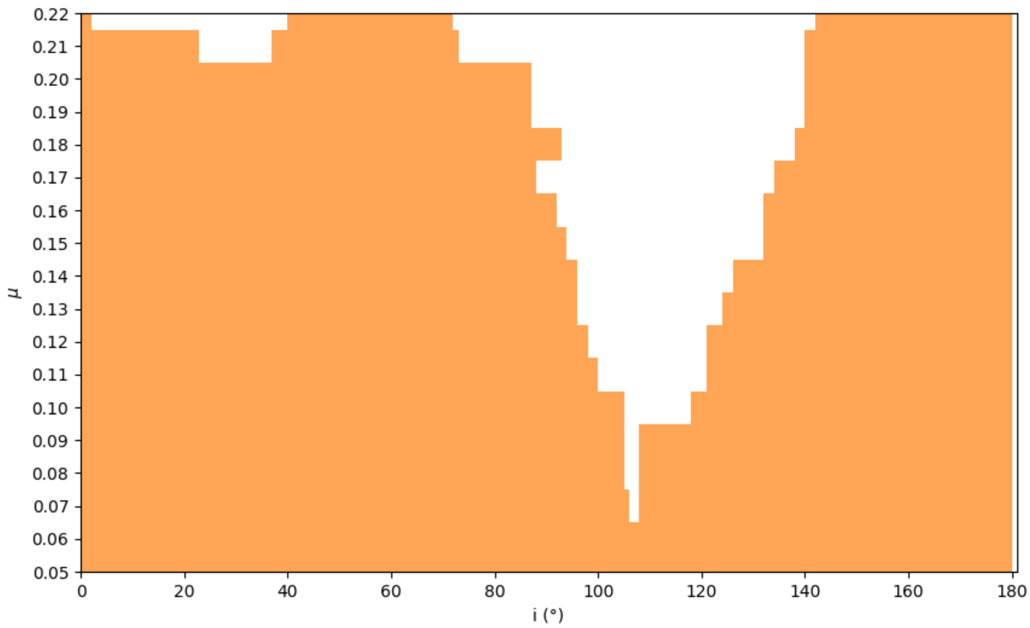


Figure 44 – Stability regions for different particle orbital inclinations. The gray areas indicate the inclination values where stable regions can be found within the range a = [0.4 - 0.7] for various values of μ.

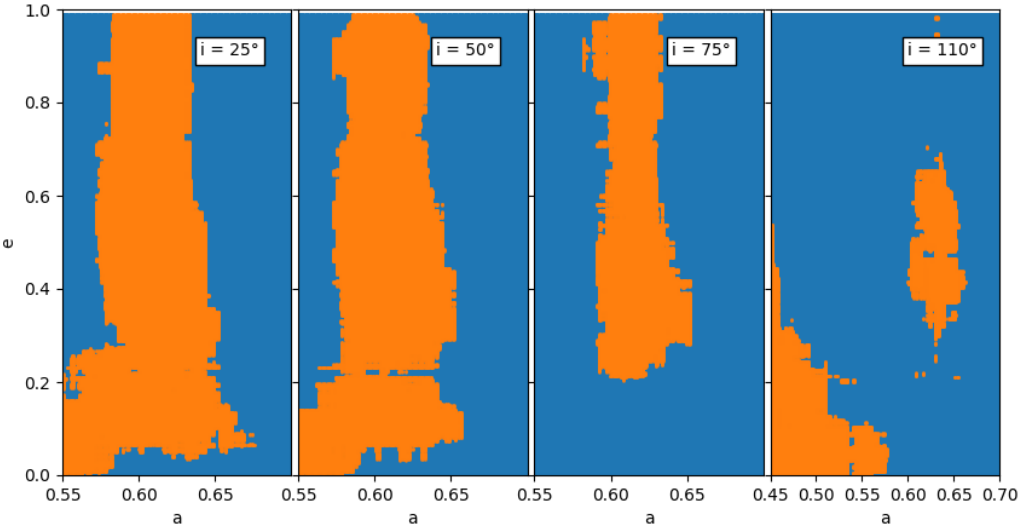
In the case of $\mu = 0.22$, the sailboat region is exceedingly tiny, such that an inclination increase of just 2° is enough to vanish all this region. For $\mu = 0.21$, the sailboat survives with an inclination up to around 20°. In both cases, a new stable region starts to appear at inclinations $\sim 35°$, extending up to approximately 70°. As inclination approaches i $\sim 140°$, a retrograde region emerges, growing until it reaches its maximum size at i $= 180°$.

For $\mu < 0.2$, the sailboat region is robust in relation to particle inclination variations and can be found for inclinations close to 90°. In μ between 0.17 and 0.2, this region can survive up to inclinations of i $= [85° - 95°]$. The sailboat region can also exist for retrograde orbits where $\mu \leq 0.16$. When the mass ratio is between 0.11 and 0.16, the region can remain stable for inclinations in the range i $= [90° - 100°]$. These results are consistent with those of Giuliatti Winter et al. (2013), who found that the sailboat island can survive in the Pluto-Charon system for inclinations up to i $= 100°$ (see their figure 4).

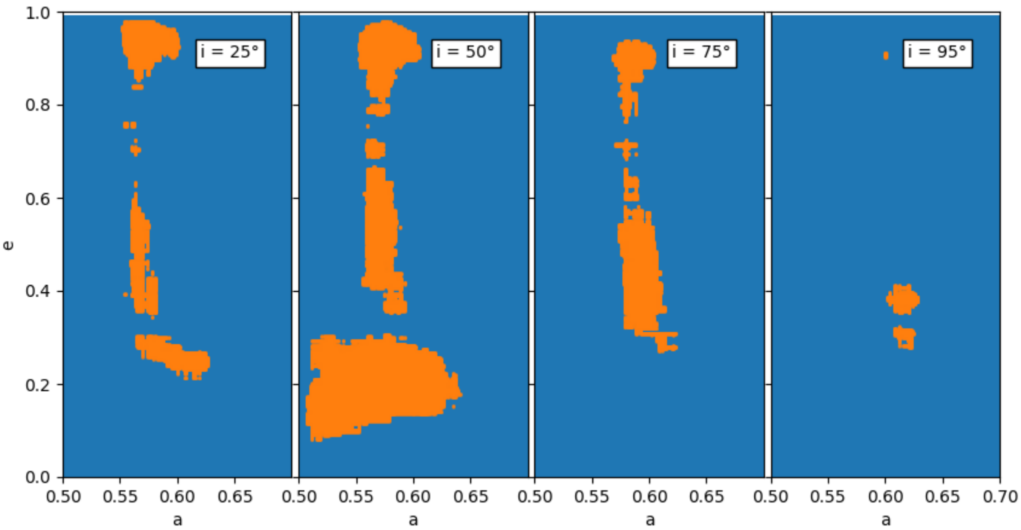
The sailboat region exists up to i $\sim 105°$ for binary mass ratios in the range $\mu =$ [0.07 - 0.1]. As the value of μ decrease, a retrograde region appears each time at lower inclinations. For $\mu =$ [0.15 - 0.2] this region surge in range of i $= [130° - 140°]$, in the case of $\mu =$ [0.11 - 0.14], this region emerges in the range of i $= [120° - 130°]$, while $\mu =$ [0.07 - 0.1], the retrograde region appears when is i $= [108° - 118°]$.

For μ are 0.06 and 0.05, the sailboat region decreases in size as inclination increases but never disappears; instead, it reaches its minimum size at a specific inclination and then expands again. In both cases of $\mu = 0.05$ and 0.06, the sailboat region reaches its minimum size at i $= 110°$.

Figure 45 shows the evolution of the sailboat for two different mass ratios, $\mu = 0.05$ and 0.12, considering particle orbital inclination are 25°, 50°, 75°, 95° (for $\mu = 0.12$) and 110° (for $\mu = 0.05$).



(a) $\mu = 0.05$



(b) $\mu = 0.12$

Figure 45 – Stability maps (a $\times$ e) showing the evolution of sailboat region (orange) varying the particle inclinations for binary systems with mass ratios of $\mu = 0.05$, 0.12.

5.6 Varying the argument of pericenter

In this analysis, we explore the effect of the particle's argument of pericenter on the sailboat region. For this purpose, we also employed a dataset containing 300,000 samples, with features including the system's mass ratio μ = [0.01 - 0.23], particle semi-major axis a = [0.4 - 0.7] , eccentricity e = [0 - 0.99] and argument of pericenter, varying from 0° to 360°.

The numerical outcomes are 11.5% of the particles survived throughout the entire simulation, 87.32% of the particles were ejected, and 1.18% detected collision between the particle and secondary body. In this dataset, our best machine learning model utilized a random oversampling technique during the training process, achieving an accuracy of 98.505%. For the stable class, the recall and precision metrics reached 94% and 93%, respectively. In the case of unstable class, both recall and precision were 99%.

Giuliatti Winter et al. (2014) demonstrated that the sailboat island exists within the Pluto-Charon system in two specific intervals of the argument of pericenter ($\Delta\omega$). The first interval is $\Delta\omega = \pm 10°$ with the maximum size at $\omega = 0°$, and the second interval is $\Delta\omega = \pm 20°$ centered around $\omega = 180°$. Our results also reveal that the sailboat region is present in two intervals of ω, reaching its maximum extension at $\omega = 0°$ and 180° independently of binary mass ratio.

In figure 46, the bars represent the approximately interval of argument of pericenter where the sailboat region exists for each μ. The plot on the left shows values of ω around 0°, and the plot on the right is around 180°.

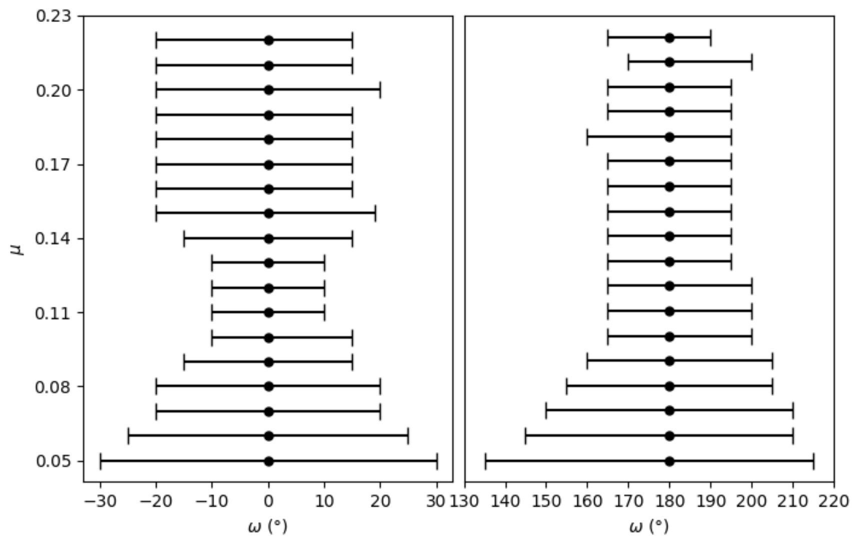


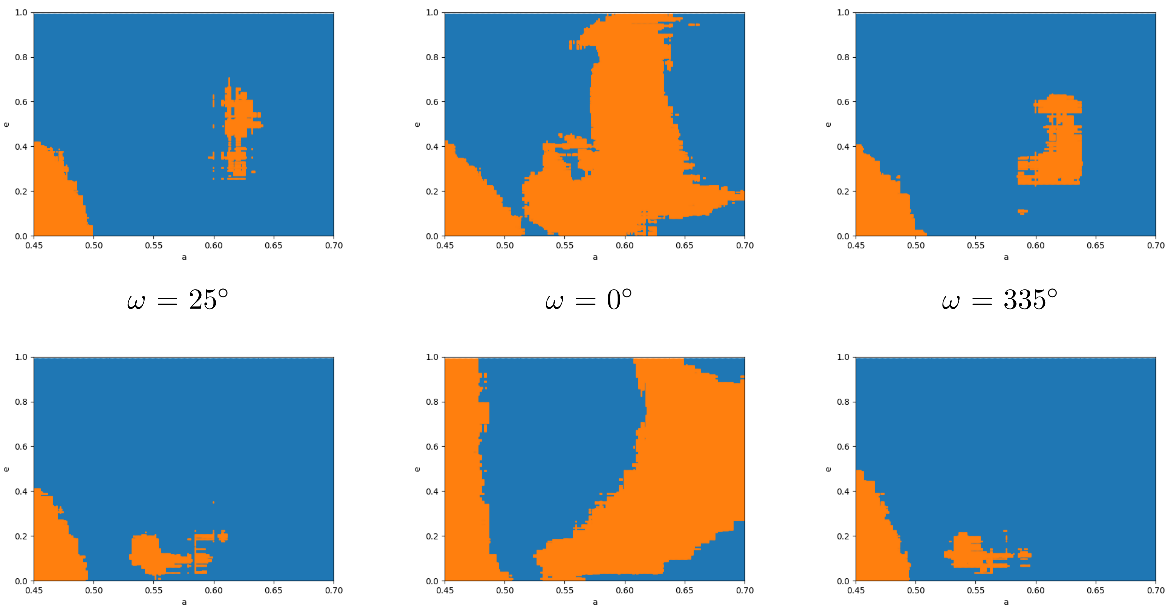
Figure 46 – The interval of particle argument of pericenter where the sailboat region exists with maximum extension at $\omega = 0°$ (left plot) and $\omega = 180°$ (right plot).

In Figure 46, for certain values of μ, the intervals of the argument of pericenter are asymmetrical around 0° or 180°, unlike the result of Giuliatti Winter et al. (2014). This difference may stem from the generalization of the algorithm, which could lead to the misclassification of some stable particles, especially as the number of these particles becomes very small in the limit of ω.

The size of these intervals does not scale proportionally with μ. For ω values around 0°, when $\mu < 0.08$ the interval $\Delta\omega \sim \pm 20°$ to $\pm 30°$. When μ is between 0.1 and 0.14, this interval narrows to approximately $\pm 10°$, matching with the results in Giuliatti Winter et al. (2014) for Pluto-Charon system. Therefore, for $\mu \geq 0.15, \Delta\omega$ increase again to about $\pm$ 20°.

Considering an interval centered at $\omega = 180°$, the sailboat region can survive within different ranges of $\Delta\omega$ depending on the value of μ. For $\mu = 0.05$, the range is $\omega = [135° - 215°]$, when μ is 0.06, the range decreases to $\omega = [145° - 210°]$. As μ increases to 0.07, the range becomes $\pm 30°$, and for $\mu = 0.08$, it further narrows to $\pm 25°$, for $\mu \leq 0.09$, the range is limited to $\omega = [160° - 205°]$. In the interval where μ = [0.10 - 0.14], the sailboat region exists within $\omega = [165° - 200°]$, for $\mu \geq 0.15$, the range is around $\pm 15°$.

Figure 47 shows the variation of sailboat region for different initial particle argument of pericenter.



$\omega = 25°$ $\omega = 0°$ $\omega = 335°$

$\omega = 140°$ $\omega = 180°$ $\omega = 210°$

Figure 47 – Diagram of initial conditions (a × e) for a binary system with $\mu = 0.05$ and different values of argument of pericenter.

5.7 Final remarks

The Sailboat is a peculiar stable region of S-type orbits around the more massive body in a binary system, located approximately midway between the separation distance of the two bodies. Its stability has also been confirmed through the Poincaré surface of section method.

In this study, we thoroughly explore the stability and constrain the range where the sailboat region exists by varying parameters such as the mass ratio of the binary system, the eccentricity of the secondary body, the particle orbital inclination, and the argument of pericenter.

We numerically simulate 1.2 million different initial conditions of elliptic three-body systems to generate four distinct datasets. These datasets are used to train a machine learning model to classify stable particles in binary systems for unknown data. In total we predicted an order of 10^9 of initial conditions with an accuracy $> 97\%$.

In the first stage, we used a dataset with three parameters: binary mass ratio, particle semi-major axis, and eccentricity. The performance for this dataset achieved an accuracy, recall, and precision of over 97% for both classes. Our results indicate that the sailboat island exists in any binary system with $\mu = [0.05 - 0.22]$. As the mass ratio of the system increases, this region shrinks in size and moves closer to the primary body. For $\mu = 0.05$, the sailboat region is located at a = [0.5 - 0.7], while for $\mu = 0.22$, the sailboat is found at a = [0.5 - 0.55] and e = [0.3 - 0.4]. Another new region emerging when $\mu =$ 0.22 and merging with the sailboat as the mass ratio slightly increases.

The second dataset contains a feature space with the parameters μ, a, e, and the eccentricity of the secondary body. The performance for this dataset achieved an accuracy of 99%, a precision and recall of 87% for stable particles, and almost 100% for unstable particles. Our findings show that slight changes in the secondary body's eccentricity are enough to make the sailboat region vanish, an empirical expression estimated the maximum eccentricity of the secondary body as a function of the binary mass ratio, demonstrating an exponential decay correlation between the two variables. For a $\mu = 0.05$ the maximum eccentricity is ~ 0.23, while when $\mu = 0.22$ the $e_p \sim 0.007$.

The third dataset includes the orbital particle parameter instead of the eccentricity of the secondary body. Its performance reached a precision and recall of 93% and 97% for the stable and unstable classes, respectively. The sailboat shows that it can survive even for retrograde orbits when $\mu \leq 0.16$, in the case of $\mu = 0.22$, only an inclination variation of 2° can sweep through the entire sailboat region, when $\mu \leq 0.6$ the sailboat diminishes in size, although it does not disappear completely.

For the fourth dataset, we studied the effects of changes in the argument of pericenter of the particle on the sailboat island. The performance of our machine learning model achieved a precision and recall of up to 94% and 99%, respectively, for the stable and unstable classes. The sailboat region exists for two different intervals of ω, which vary in size for different μ values, with its maximum extension at $\omega = 0°$ and $180°$.

6 Final remarks

This text presents three different projects that apply machine learning techniques to classify stable particles orbiting one of the two massive bodies in the elliptic three-body problem.

In Chapter 2, we reviewed key concepts in classification learning, focusing on five tree-based algorithms: decision tree, random forest, XGBoost, LightGBM, and histogram-based gradient boosting. Additionally, we explored methods to handle with imbalanced datasets and strategies to search the best-performance of machine learning models.

In the Chapter 3, we demonstrated the effectiveness of machine learning in predicting orbital stability planet maps. In this work, we used a dataset of 100,000 simulations of the elliptic three-body problem, where each numerical simulations has nine different initial parameters, representing the orbital element of the planet and the particle. The XGBoost achieved the highest accuracy of 98.48% and this model successfully reproduced the results of prior studies such as Domingos et al. (2006) and Pinheiro and Sfair (2021) as well the stability map of Saturn's Inuit group of satellites. Machine learning significantly reduced computational time compared to traditional methods, and future work aims to improve model performance at stable boundaries regions, we also plan to make our model available through a public web interface.

The Chapter 4 investigated the mass, eccentricity, ring size, and inclination of two exoring candidates, J1407b and PDS110b, using machine learning to predict stable particles in these systems. We used the best-performance model from Chapter 3 to predicted 2.7 billion different initial conditions, the stability maps allowed for estimates of the maximum ring size and eclipse duration. In the case of PDS110b, the model suggests that a planet with a mass between 15.2 and 16.9 M_{Jup} in a near-circular orbit ($e_p < 0.1$) provides the highest probability (7.2%) to reproduce the observable eclipses in 2008 and 2011. The ring's radius is estimated to be between 0.2 and 0.3 au, with an inclination $\geq 150°$.

In the case of possible exoring system around J1407b, the dimming event is explained by a planet with a semi-major axis between 3 and 15 au, with a probability < 36% and if the semi-major axis exceeds 5 au and the planet has a high eccentricity ($ep > 0.7$). However, our findings showed that a planet with mass < 9.5 M_{Jup}. cannot reproduce the observable eclipse with a transverse velocity > 9 kms^{-1}, which not match with the transversal velocity of 32 ± 2 kms^{-1} estimated by (Kenworthy et al., 2015).

The Chapter 5 studied the stability of a peculiar stable region in binary systems, referred to as the sailboat region, for various initial parameters such as the mass ratio of the system, the eccentricity of the secondary body, the particle's orbital inclination, and the argument of pericenter. We generated four distinct datasets to train machine learning models, which classified stable particles with an accuracy exceeding 97%. The results suggest that the sailboat region exists in binary systems with mass ratios μ = [0.05 - 0.22].

The sailboat region is highly sensitive to changes in the secondary body's eccentricity, showing an exponential decay when comparing the maximum eccentricity of the secondary body with μ. Regarding particle inclination, the sailboat region can also exist for retrograde orbits when $\mu \leq 0.16$. Additionally, the argument of pericenter of the particles also affects the extension of the sailboat region. It survives within two intervals of ω, and its maximum size always occurs when ω is $0°$ or $180°$.

Overall, machine learning models showed a significant reduction in computational time, enabling the exploration of large initial parameter spaces while achieving highly accurate performance.

References

M. Aizawa, S. Uehara, K. Masuda, H. Kawahara, and Y. Suto. Toward detection of exoplanetary rings via transit photometry: Methodology and a possible candidate. *The Astronomical Journal*, 153(4):193, 2017.

M. Aizawa, K. Masuda, H. Kawahara, and Y. Suto. Systematic search for rings around kepler planet candidates: Constraints on ring size and occurrence rate. *The Astronomical Journal*, 155(5):206, 2018.

E. Ashton, B. Gladman, and M. Beaudoin. Evidence for a recent collision in saturn's irregular moon population. *The Planetary Science Journal*, 2(4):158, 2021.

J. W. Barnes and J. J. Fortney. Transit detectability of ring systems around extrasolar giant planets. *The Astrophysical Journal*, 616(2):1193, 2004.

C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

I. Bondarenko and E. Perevozkina. Vizier online data catalog: Catalog of early-type contact binaries (bondarenko+, 1996). *VizieR Online Data Catalog*, pages V–120, 2004.

F. Braga-Ribas, B. Sicardy, J. Ortiz, C. Snodgrass, F. Roques, R. Vieira-Martins, J. Camargo, M. Assafin, R. Duffard, E. Jehin, et al. A ring system detected around the centaur (10199) chariklo. *Nature*, 508(7494):72–75, 2014.

L. Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

R. A. Broucke. Periodic orbits in the restricted three-body problem with earth-moon masses. Technical Report 32-1168, Jet Propultion Laboratory, California Institute of Technoly, 1968.

M. Brown, D. Ragozzine, J. Stansberry, and W. Fraser. The size, density, and formation of the orcus–vanth system in the kuiper belt. *The Astronomical Journal*, 139(6):2700, 2010.

M. E. Brown and B. J. Butler. Masses and densities of dwarf planet satellites measured with alma. *The Planetary Science Journal*, 4(10):193, 2023.

M. Brozović and R. A. Jacobson. Post-new-horizons orbits and masses for the satellites of pluto. *The Astronomical Journal*, 167(6):256, 2024.

V. Carruba, S. Aljbaae, R. C. Domingos, and W. Barletta. Artificial neural network classification of asteroids in the m1: 2 mean-motion resonance with mars. *Monthly Notices of the Royal Astronomical Society*, 504(1):692–700, 2021.

V. Carruba, S. Aljbaae, R. Domingos, M. Huaman, and W. Barletta. Machine learning applied to asteroid dynamics. *Celestial Mechanics and Dynamical Astronomy*, 134(4): 36, 2022.

V. Carruba, S. Aljbaae, G. Caritá, M. Lourenço, B. Martins, and A. Alves. Imbalanced classification applied to asteroid resonant dynamics. *Frontiers in Astronomy and Space Sciences*, 10:1196223, 2023.

V. Carruba, J. I. Camargo, S. Aljbaae, F. Ferreira, E. Lin, V. Figueiredo-Peixoto, M. Banda-Huarca, A. Pieres, R. Boufleur, L. Da Costa, et al. Main belt asteroids taxonomical information from dark energy survey data. *Monthly Notices of the Royal Astronomical Society*, 527(3):6495–6505, 2024.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

R. Domingos, O. Winter, and T. Yokoyama. Stable satellites around extrasolar giant planets. *Monthly Notices of the Royal Astronomical Society*, 373(3):1227–1234, 2006.

R. Dvorak. Critical orbits in the elliptic restricted three-body problem. *Astronomy and Astrophysics (ISSN 0004-6361), vol. 167, Oct. 1986, p. 379-386.*, 167:379–386, 1986.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

T. Fushiki. Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21:137–146, 2011.

A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc.", 2022.

S. Giuliatti Winter, O. Winter, E. Vieira Neto, and R. Sfair. Stable regions around pluto. *Monthly Notices of the Royal Astronomical Society*, 430(3):1892–1900, 2013.

S. M. Giuliatti Winter, O. Winter, E. Vieira Neto, and R. Sfair. A peculiar stable region around pluto. *Monthly Notices of the Royal Astronomical Society*, 439(4):3300–3307, 2014.

T. Grav and J. Bauer. A deeper look at the colors of the saturnian irregular satellites. *Icarus*, 191(1):267–285, 2007.

W. Grundy, K. Noll, H. Roe, M. Buie, S. Porter, A. Parker, D. Nesvornỳ, H. Levison, S. Benecchi, D. Stephens, et al. Mutual orbit orientations of transneptunian binaries. *Icarus*, 334:62–78, 2019.

W. M. Grundy, S. B. Porter, S. D. Benecchi, H. G. Roe, K. S. Noll, C. A. Trujillo, A. Thirouin, J. A. Stansberry, E. Barker, and H. F. Levison. The mutual orbit, mass, and density of the large transneptunian binary system varda and ilmarë. *Icarus*, 257: 130–138, 2015.

H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. Ieee, 2008.

M. Z. Heising, G. W. Marcy, and H. E. Schlichting. A search for ringed exoplanets using kepler photometry. *The Astrophysical Journal*, 814(1):81, 2015.

R. Heller. On the detection of extrasolar moons and rings. *arXiv preprint arXiv:1701.04706*, 2017.

B. Holler, W. Grundy, M. Buie, and K. Noll. The eris/dysnomia system i: The orbit of dysnomia. *Icarus*, 355:114130, Feb. 2021. ISSN 0019-1035. doi: 10.1016/j.icarus.2020.114130. URL http://dx.doi.org/10.1016/j.icarus.2020.114130.

M. J. Holman and P. A. Wiegert. Long-term stability of planets in binary systems. *The Astronomical Journal*, 117(1):621, 1999.

R. Hunter. Motions of satellites and asteroids under the influence of jupiter and the sun: I. stable and unstable satellite orbits. *Monthly Notices of the Royal Astronomical Society*, 136(3):245–265, 1967.

M. Ibrahim, H. AbdelRaouf, K. M. Amin, and N. Semary. Keystroke dynamics based user authentication using histogram gradient boosting. *IJCI. International Journal of Computers and Information*, 10(1):36–53, 2023.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

M. Kenworthy, S. Lacour, A. Kraus, A. Triaud, E. E. Mamajek, E. Scott, et al. Mass and period limits on the ringed companion transiting the young star j1407. *Monthly Notices of the Royal Astronomical Society*, 446(1):411–427, 2015.

M. A. Kenworthy and E. E. Mamajek. Modeling giant extrasolar ring systems in eclipse and the case of j1407b sculpting by exomoons? *The Astrophysical Journal*, 800(2):126, 2015.

C. Kiss, G. Marton, A. H. Parker, W. M. Grundy, A. Farkas-Takács, J. Stansberry, A. Pal, T. Müller, K. S. Noll, M. E. Schwamb, et al. The mass and density of the dwarf planet (225088) 2007 or10. *Icarus*, 334:3–10, 2019.

Z. Kopal. The classification of close binary systems. *Annales d'Astrophysique, Vol. 18, p. 379*, 18:379, 1955.

M. Kumar and H. Sheshadri. On the classification of imbalanced datasets. *International Journal of Computer Applications*, 44(8):1–7, 2012.

C. Lam and D. Kipping. A machine learns to predict the stability of circumbinary planets. *Monthly Notices of the Royal Astronomical Society*, 476(4):5692–5697, 2018.

M. Learning. Tom mitchell. *Publisher: McGraw Hill*, 1997.

P. Liashchynskyi and P. Liashchynskyi. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.

E. E. Mamajek, A. C. Quillen, M. J. Pecaut, F. Moolekamp, E. L. Scott, M. A. Kenworthy, A. C. Cameron, and N. R. Parley. Planetary construction zones in occultation: Discovery of an extrasolar ring system transiting a young sun-like star and future prospects for detecting eclipses by circumsecondary and circumplanetary disks. *The Astronomical Journal*, 143(3):72, 2012.

M. Melita and J. Papaloizou. Apse-alignment in narrow-eccentric ringlets and its implications for the ϵ-ring of uranus and the ring system of (10199) chariklo. *Icarus*, 335: 113366, 2020. ISSN 0019-1035.

R. Mentel, M. Kenworthy, D. Cameron, E. Scott, S. Mellon, R. Hudec, J. Birkby, E. Mamajek, A. Schrimpf, D. Reichart, et al. Constraining the period of the ringed secondary companion to the young star j1407 with photographic plates. *Astronomy & Astrophysics*, 619:A157, 2018.

R. Mohammed, J. Rawashdeh, and M. Abdullah. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*, pages 243–248. IEEE, 2020.

B. Morgado, B. Sicardy, F. Braga-Ribas, J. Ortiz, H. Salo, F. Vachier, J. Desmars, C. Pereira, P. Santos-Sanz, R. Sfair, et al. A dense ring of the trans-neptunian object quaoar outside its roche limit. *Nature*, 614(7947):239–243, 2023.

C. D. Murray and S. F. Dermott. *Solar system dynamics*. Cambridge university press, 1999.

R. Narimani, C. Jun, A. Saedi, S. M. Bateni, and J. Oh. A multivariate decomposition–ensemble model for estimating long-term rainfall dynamics. *Climate Dynamics*, 61(3): 1625–1641, 2023.

E. V. Neto and O. Winter. Time analysis for temporary gravitational capture: satellites of uranus. *The Astronomical Journal*, 122(1):440, 2001.

Y. Ohta, A. Taruya, and Y. Suto. Predicting photometric and spectroscopic signatures of rings around transiting extrasolar planets. *The Astrophysical Journal*, 690(1):1, 2008.

J. L. Ortiz, P. Santos-Sanz, B. Sicardy, G. Benedetti-Rossi, D. Bérard, N. Morales, R. Duffard, F. Braga-Ribas, U. Hopp, C. Ries, et al. The size, shape, density and ring of the dwarf planet haumea from a stellar occultation. *Nature*, 550(7675):219–223, 2017.

H. Osborn, J. Rodriguez, M. Kenworthy, G. Kennedy, E. Mamajek, C. Robinson, C. Espaillat, D. Armstrong, B. Shappee, A. Bieryla, et al. Periodic eclipses of the young star pds 110 discovered with wasp and kelt photometry. *Monthly Notices of the Royal Astronomical Society*, 471(1):740–749, 2017.

H. P. Osborn, M. Kenworthy, J. E. Rodriguez, E. J. de Mooij, G. M. Kennedy, H. Relles, E. Gomez, M. Hippke, M. Banfi, L. Barbieri, et al. The pds 110 observing campaign–photometric and spectroscopic observations reveal eclipses are aperiodic. *Monthly Notices of the Royal Astronomical Society*, 485(2):1614–1625, 2019.

C. Pereira, B. Sicardy, B. Morgado, F. Braga-Ribas, E. Fernández-Valenzuela, D. Souami, B. Holler, R. Boufleur, G. Margoti, M. Assafin, et al. The two rings of (50000) quaoar. *Astronomy & Astrophysics*, 673:L4, 2023.

T. F. Pinheiro and R. Sfair. Constraining the nature of the possible extrasolar pds110b ring system. *Astronomy & Astrophysics*, 652:A149, 2021.

J. R. Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

D. Ragozzine and M. E. Brown. Orbits and masses of the satellites of the dwarf planet haumea (2003 el61). *The Astronomical Journal*, 137(6):4766, 2009.

H. Rein and D. S. Spiegel. ias15: a fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits. *Monthly Notices of the Royal Astronomical Society*, 446(2):1424–1437, 11 2014. ISSN 0035-8711.

H. Rein and D. S. Spiegel. Ias15: a fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits. *Monthly Notices of the Royal Astronomical Society*, 446(2):1424–1437, 2015.

S. Rieder and M. A. Kenworthy. Constraints on the size and dynamics of the j1407b ring system. *Astronomy & Astrophysics*, 596:A9, 2016.

O. Sagi and L. Rokach. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 8(4):e1249, 2018.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

J. Schneider, C. Dedieu, P. Le Sidaner, R. Savalle, and I. Zolotukhin. Defining and cataloging exoplanets: the exoplanet. eu database. *Astronomy & Astrophysics*, 532:A79, 2011.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014a.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014b.

S. S. Sheppard. The color differences of kuiper belt objects in resonance with neptune. *The Astronomical Journal*, 144(6):169, 2012.

S. S. Sheppard, Y. R. Fernandez, and A. Moullet. The albedos, sizes, colors, and satellites of dwarf planets compared with newly measured dwarf planet 2013 fy27. *The Astronomical Journal*, 156(6):270, 2018.

S. S. Sheppard, D. J. Tholen, M. Alexandersen, and C. A. Trujillo. New jupiter and saturn satellites reveal new moon dynamical families. *Research Notes of the AAS*, 7(5):100, 2023.

J. Speedie and J. Zanazzi. The structure and stability of extended, inclined circumplanetary disc or ring systems. *Monthly Notices of the Royal Astronomical Society*, 497(2):1870–1883, 2020.

S. V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89, 1997.

L. Surkova and M. Svechnikov. Vizier online data catalog: Semi-detached eclipsing binaries (surkova+, 2004). *VizieR Online Data Catalog*, 2004.

P. J. Sutton, B. Albery, and J. Muff. A self-gravitating exoring around j1407b and implications for in-situ exomoon formation. *Frontiers in Astronomy and Space Sciences*, 9:819933, 2022.

R. S. Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

M. Svechnikov and E. Perevozkina. Vizier online data catalog: Catalog of dms-type eclipsing binaries (svechnikov+, 1999). *VizieR Online Data Catalog*, page 5, 2004.

D. Tamayo, A. Silburt, D. Valencia, K. Menou, M. Ali-Dib, C. Petrovich, C. X. Huang, H. Rein, C. Van Laerhoven, A. Paradise, et al. A machine learns to predict the stability of tightly packed planetary systems. *The Astrophysical Journal Letters*, 832(2):L22, 2016.

L. R. M. Tusnski and A. Valio. Transit model of planets with moon and ring systems. *The Astrophysical Journal*, 743(1):97, 2011.

T. Van Werkhoven, M. Kenworthy, and E. Mamajek. Analysis of 1swasp j140747. 93-394542.6 eclipse fine-structure: hints of exomoons. *Monthly Notices of the Royal Astronomical Society*, 441(4):2845–2854, 2014.

C. Wade and K. Glynn. *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python.* Packt Publishing Ltd, 2020.

H. J. Weerts, A. C. Mueller, and J. Vanschoren. Importance of tuning hyperparameters of machine learning algorithms. *arXiv preprint arXiv:2007.07588*, 2020.

S. M. G. Winter, O. C. Winter, E. V. Neto, and R. Sfair. The sailboat island and the new horizons trajectory. *Icarus*, 246:339–344, 2015.

Z. Zheng, Y. Cai, and Y. Li. Oversampling method for imbalanced classification. *Computing and Informatics*, 34(5):1017–1037, 2015.

Q. Zou, S. Xie, Z. Lin, M. Wu, and Y. Ju. Finding the best classification threshold in imbalanced classification. *Big Data Research*, 5:2–8, 2016.

APPENDIX A – PDS110b ring system

Copy of the published version of the paper 'Constraining the nature of the possible extrasolar PDS110b ring system' in the journal Astronomy & Astrophysics, 652.

A&A 652, A149 (2021)
https://doi.org/10.1051/0004-6361/202039556

Astronomy & Astrophysics

Constraining the nature of the possible extrasolar PDS110b ring system

Tiago F. L. L. Pinheiro and Rafael Sfair

Grupo de Dinâmica Orbital e Planetologia, São Paulo State University, UNESP, Guaratinguetá, CEP 12516-410, São Paulo, Brazil
e-mail: francisco.pinheiro@unesp.br; rafael.sfair@unesp.br

Received 29 September 2020 / Accepted 17 June 2021

ABSTRACT

The young star PDS 110 in the Ori OB 1a association underwent two similar eclipses in 2008 and 2011, each of which lasted for a period of at least 25 days. One plausible explanation for these events is that the star was eclipsed by an unseen giant planet (named PDS 110b) circled by a ring system that fills a large fraction of its Hill sphere. Through thousands of numerical simulations of the three-body problem, we constrain the mass and eccentricity of this planet as well the size and inclination of its ring, parameters that are not well determined by the observational data alone. We carried out a broad range of different configurations for the PDS110b ring system and ruled out all that did not match with the observations. The result shows that the ring system could be prograde or retrograde; the preferred solution is that the ring has an inclination lower than 60° and a radius between 0.1 and 0.2 au and that the planet is more massive than 35 M_{Jup} and has a low eccentricity (<0.05).

Key words. planets and satellites: rings – planets and satellites: dynamical evolution and stability – eclipses

1. Introduction

The first exoplanet discovered was around a pulsar, PSR1257+12 (Wolszczan & Frail 1992), and three years later the first detection of a planet (51 Pegasi b) orbiting a solar-type star occurred (Mayor & Queloz 1995). In our Solar System, all four giant planets host ring systems; however, as of now, no extrasolar ring system (exoring) has been detected, and the detection of a Saturn-like ring system requires high time-resolution photometry (Barnes & Fortney 2004). The most useful method for detecting the presence of a ring is the planetary transit (Akinsanmi et al. 2020), and several works have searched for signatures of a planetary ring system around extrasolar Kepler planet candidates (e.g., Heising et al. 2015, and Aizawa et al. 2017, 2018).

Schlichting & Chang (2011) analyzed the nature of possible ring systems that could exist around an exoplanet with an orbital period of about one year or less. Their results show that 24% of the evaluated planets have the necessary conditions to support sizable rings, and, since these planets would be very close to their host star, the orbiting disk around them probably consists of rocky material.

The first exoring candidate was reported in 2007, when the star J1407 underwent a complex series of eclipses that lasted about 56 days and achieved a depth greater than three magnitudes (Mamajek et al. 2012). One potential solution for this event is the passage of a giant ring that extends out to a radius of 0.6 au, surrounding an unseen secondary companion that is in front of the star. The structure has a gap at 0.4 au, which, according to Kenworthy & Mamajek (2015), could be a result of an exomoon clearing out this region. This exosatellite is expected to have a mass comparable to that of the Earth. Rieder & Kenworthy (2016) investigated, through numerical simulation, the effect on the stability of the ring system around J1407b in an eccentric orbit ($0.6 \leq e \leq 0.7$) and with an orbital period of 11 yr. The results strongly suggest that a retrograde ring and a secondary companion with a mass of 60–100 M_{Jup} can fit the observable eclipse.

Speedie & Zanazzi (2020) investigated the stability and structure of a circumplanetary ring system around an oblate and oblique planet under the influence of the spinning planet's mass quadrupole coefficient (J_2) through N-body integrations. Their results show that a massive planet (10 M_{Jup}) orbiting a Sun-like star either in circular or eccentric orbit can wrap a stable ring that extends out to a 2 Laplace radius. The authors identified two main instability mechanisms. The first is the Lidov-Kozai instability, which occurs for a planet with an obliquity of 80°, and the second is ivection instability. Even though there is no measurement of the oblateness coefficient, they assumed $J_2 = 0.5$ for J1407b (30 times larger than Saturn's), finding that the likely mass for the planet is greater than 13 M_{Jup}.

Another possible exoring structure was found in the PDS 110 system. The young star PDS 110 (∼11 Myr old), which is roughly two times more massive than the Sun, underwent two series of eclipses in 2008 and 2011, with each dimming event lasting about 25 days with a depth of 30%. The infrared excess in the total luminosity of PDS 110 indicates the presence of a dust disk encircling the star, and the similarity between the events suggests the presence of a periodic unseen secondary companion surrounded by a circum-secondary disk structure (Osborn & Rodriguez 2017). Another eclipse was predicted to happen in 2017, and a ground-based observing campaign was organized to monitor the event. However, no drop greater than 1% was detected. Osborn & Kenworthy (2019) propose that, instead of a ring, a circumstellar dust disk around PDS 110 could be the source of variability. Future observations can confirm the existence of this ring system and the secondary companion.

A&A 652, A149 (2021)

A number of characteristics of the secondary companion could not be well determined from the observations. Osborn & Rodriguez (2017) reported that the hypothetical companion orbits the PDS110 star at 2 au and that its mass is in the range of [1.8–70] $M_{\rm Jup}$, which would indicate it is a planet or a brown dwarf. The presumed ring would have a radius of about 0.15 au.

The goal of this work is to analyze the stability of a putative exoring system around an eccentric, massive planet at 2 au from the PDS110 star that matches with the eclipses that occurred in 2008 and 2011. Given the extensive range of possible values for a planet's mass and eccentricity, ring size, and inclination, we aim to restrict the physical and orbital parameters of this system through numerical simulation.

In Sect. 2 we describe in detail the ring system and the two main eclipses. In Sect. 3 we present the first constraints on the upper limit for the eccentricity of the planet, while in Sect. 4 we report the numerical model for the putative systems, exploring different values for the mass and eccentricity of the planet and for the orbital inclination of the ring. The results of the numerical simulations are discussed in Sect. 5, for both prograde and retrograde cases. Our final remarks are presented in the last section.

2. The PDS110b ring system

The star PDS110, also identified as IRAS 05209-0107 or HD 290380, is an ~11 Myr old GE/FE-type star and a member of the Ori OB1a association at a distance of 345 ± 40 pc (Brown et al. 2016). It has a radius roughly equivalent to the diameter of the Sun (2.23 $R_\odot$) and a mass ~1.6 times the solar mass. The presence of infrared excess, approximately 25% of the total luminosity ($L_{\rm IR}/L_{\rm bol} = 0.25$; Osborn & Rodriguez 2017), indicates the presence of two protoplanetary disks surrounding the star: an inner one with an extension of approximately 0.2 au, too small to be directly detected by images, and a second inclined outer disk, formed by dust with an extension of up to 300 au.

The PDS110 system underwent two similar dimming events, each over 25 days and with a brightness drop of ~30%. The events were observed in 2008 and 2011, as reported by Osborn & Rodriguez (2017) and Osborn & Kenworthy (2019). An analysis by Osborn & Rodriguez (2017) investigated two different interpretations for the eclipses. The first possible explanation was the passage of an inclined disk around an unseen low mass secondary companion, possibly a brown dwarf or a giant planet. The authors concluded that any large disk around PDS110 would have been detected in the optical spectra, as would any indication of a secondary star in the system. The second and more probable explanation for the eclipses is that a circumplanetary disk around PDS110b caused them.

The interval of two years between eclipses suggests an orbital period for the secondary companion of 808 days. Considering that the mass of the star is 1.6 $M_\odot$, the estimated semimajor axis of PDS110b would be ~2 au. An analytical analysis of the eclipse made by Osborn & Rodriguez (2017), assuming a circular orbit for the planet, implies that the secondary companion may have a mass between 1.8 and 70 $M_{\rm Jup}$. Furthermore, from the orbital speed and eclipse duration, and assuming the secondary has a circular orbit, the authors estimated the disk diameter to be around 0.3 au. Based on the observational data, no constraint on the planet eccentricity was made.

Here we explore the case that the secondary body is a planet (hereafter called PDS110b) that hosts an extensive ring system that was invisible during the eclipses. Our goal is to better constrain the physical and orbital parameters of the PDS110b system through numerical simulations and some analytical analysis in order to explain the dimming events.

3. Constraints on the eccentricity of PDS110b

Initially, we performed an analytic analysis to constrain an upper limit for the eccentricity of PDS110b by correlating the ring orbital radius with the Hill region of the planet.

The Hill radius ($r_{\rm Hill}$) is the distance from the planet where the gravitational influence of the host planet dominates over the stellar perturbation. In the case of an eccentric orbit, the Hill radius can be estimated as (Hamilton & Burns 1992)

$$r_{\rm Hill} \approx a(1-e)\sqrt[3]{\frac{m_{\rm p}}{3M_{\rm s}}}, \tag{1}$$

where a is the semimajor axis, e is the eccentricity, and $m_{\rm p}$ and $M_{\rm s}$ are the masses of the planet and the star, respectively.

Since the disk inclination and the planet obliquity are unknown, we assume the ring inclination observed from Earth is 0° (face-on) and that the ring is in the equatorial plane of the planet. Now let $t_{\rm e}$ be the eclipse duration time, assuming the planet travels in a straight line with a constant velocity $v_{\rm e}$. The radial size of the projected ring, $r_{\rm ring}$, can be written as

$$r_{\rm ring} = \frac{t_{\rm e} v_{\rm e}}{2}, \tag{2}$$

and the two parameters can be combined to define ξ as the ratio between the ring and the Hill radius

$$\xi = \frac{r_{\rm ring}}{r_{\rm Hill}}, \tag{3}$$

which helps us understand the stability of the ring.

We considered the mass of the star to be $M_{\rm s} = 1.6\ M_\odot$, the observed period of the planet to be $P = 808$ days, and the eclipse to last $t_{\rm e} = 25$ days. To explore an extreme configuration, we assumed the ring to be in the orbital plane of the planet and the eclipse to happen during the passage of the planet through the apocenter (when the planet velocity is at its minimum). With these assumptions, ξ was calculated for the secondary companion mass in the range of [1.8–70] $M_{\rm Jup}$ (Osborn & Rodriguez 2017) and for different values of eccentricity.

If $\xi < 1$, the ring is within the gravitational domain of the planet. Hunter (1967) studied the stability orbits of hypothetical satellites around Jupiter disturbed by the Sun, with different values for the semimajor axis, eccentricity, and inclination. Their results showed that the stability region for prograde orbits is smaller than 0.45 $r_{\rm Hill}$, while for retrograde orbits the extension of this region is limited to $0.75 r_{\rm Hill}$. Domingos et al. (2006) studied the stability zone of satellites around extrasolar giant planets and verified that the limit of the stable region is ~0.5 $r_{\rm Hill}$ for prograde orbits and around ~0.95 $r_{\rm Hill}$ for the retrograde case (they restricted the analysis to a mass ratio between the massive bodies of a maximum of 10^{-3}). Therefore, in our analysis we consider systems to be stable when $\xi < 0.5$.

Figure 1 shows ξ as a function of the mass of the planet for different values of the planet's eccentricity ($e_{\rm p}$). The horizontal line corresponds to the stability limit of $\xi = 0.5$, allowing us to constrain the mass of the planet to be greater than 35 $M_{\rm Jup}$ and

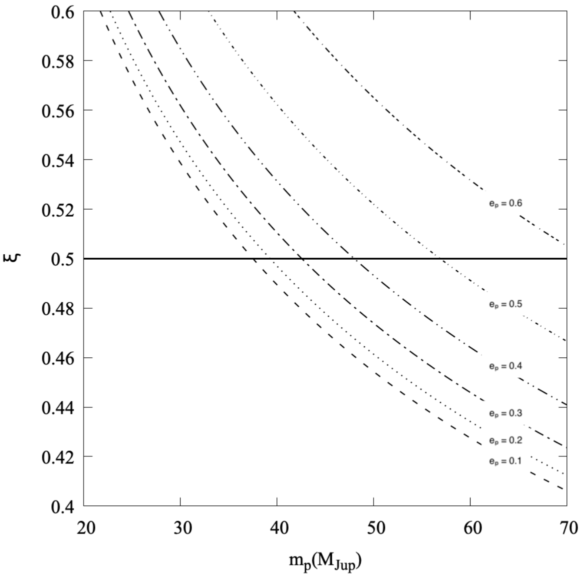


Fig. 1. Factor ξ in terms of planet mass (m_p) and its eccentricity (e_p). The line at $\xi = 0.5$ delineates the stability zone (which is below this line).

to rule out systems where the eccentricity of the planet is larger than 0.5.

The high eccentricity implies that the planet transit velocity is greater than 15 km s^{-1}, which is consistent with the minimal eclipse velocity detected by Osborn & Rodriguez (2017) through observational data. From the flux increase in the light curve, they calculated the minimum velocity of the object during the event to be $v_{min} = 13$ km s^{-1}.

4. Numerical method

To investigate the possible physical and orbital parameters of the PDS110b ring system, we numerically modeled the system as an ensemble of three-body problems: the star, a planet, and the ring particle, where each numerical simulation represents one different initial condition of the system. In total, we ran 1.3×10^6 different simulations.

For all models, we set the values of PDS110 and its companion, PDS110b, to be in agreement with the values from Osborn & Rodriguez (2017). The mass of the star is 1.6 $M_\odot$, and its radius is 2.23 $M_\odot$. The parameters of the hypothetical planet were assumed considering the following grid of parameters: The semimajor axis (a_p) is fixed to 2 au, while the eccentricity (e_p) and the mass (m_p) are randomly chosen within the ranges of 0 to 0.5 and 1.8–70 M_{Jup}, respectively. These intervals were determined based on the references presented in Sect. 2; however, for the sake of completeness, we decided to analyze the entire mass range proposed in Osborn & Rodriguez (2017). The radius of the planet (r_p) was calculated assuming a density of 1.326 kg m^{-3} (the same as Jupiter's).

We considered the ring particle as a test particle orbiting the planet and gravitationally perturbed by the star. The particle is initially in a circular orbit, and the semimajor axis (a) is chosen randomly from the range 0.01 to 0.5 au and the true anomaly (f) from 0° to 360°. We assumed the other parameters of the particles, the longitude of the ascending node (Ω) and the argument of pericenter (ω), to be 0°. Assuming the possibility that the eclipses occurred via an inclined ring

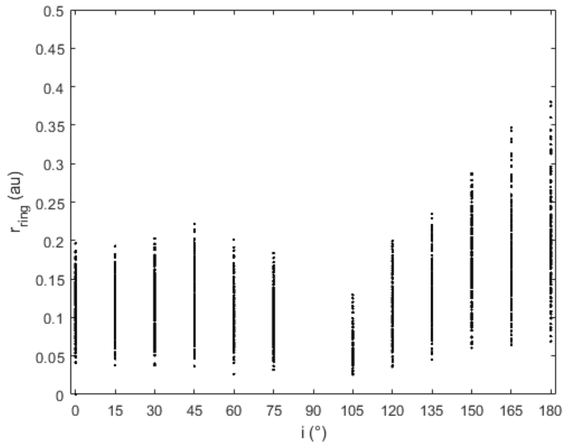


Fig. 2. Ring radius of each group of simulations based on a ring inclination from $i = [0°$ to $180°]$.

(Osborn & Rodriguez 2017), we varied, for each set of 10^5 numerical simulations, the inclination of the particle (i), defined as the angle between the ring plane and the orbital planet of the planet, from 0° to 180° in steps of $\Delta i = 15°$.

The time span for all numerical simulations was set as 250×10^4 days, the equivalent of 10^4 orbital periods of the particle with a semimajor axis a = 0.5 au and orbiting a planet with a mass of 1.8 M_{Jup}. Among all the possible scenarios, this is the largest period that one particle could have. The numerical integrations were carried out with the Rebound package using the IAS15 integrator (Rein & Spiegel 2014).

Whenever the particle collided with one of the massive bodies or was ejected from the system, the simulation was interrupted. The collisions were defined when the distance between the particle and a massive body (star or planet) was smaller than the body's radius, and particles were considered ejected when their orbit became hyperbolic ($e < 1$) in reference to the planet. All the particles that survived the entire time of integration were considered to be stable.

We divided our set of simulations into 1820 different groups according to the inclination of the ring particle, the eccentricity, and the mass of the planet, by grouping them within the following intervals: $\Delta i = 15°$, $\Delta e_p = 0.05$, and $\Delta m_p \approx 5\ M_{Jup}$. For each set, we determined the radial distribution of the surviving particles throughout the entire integration. To remove outliers, the ring extension (r_{ring}) in each group was defined as the semimajor axis distance from the farthest particle within the area where the nearest 98% of the ensemble remained.

5. Numerical results

Given the large number of variables, we explored thousands of different sets of initial conditions within the ranges described in Sect. 4. For the complete ensemble, the overall outcome was as follows: 6.3% of the cases ended in collision and 72.2% ended in escape, while in the remaining 21.5% of the cases the particles remained stable throughout the entire integration. It is interesting to note that the majority of the unstable particles (~93%) were ejected or collided with one of the massive bodies in less than 40 orbital periods of the planet (<80 yr).

We also verified the relation between the orbital inclination of the particle and the ring radius (Fig. 2). Each dot represents a group of simulations. We see that the ring extension

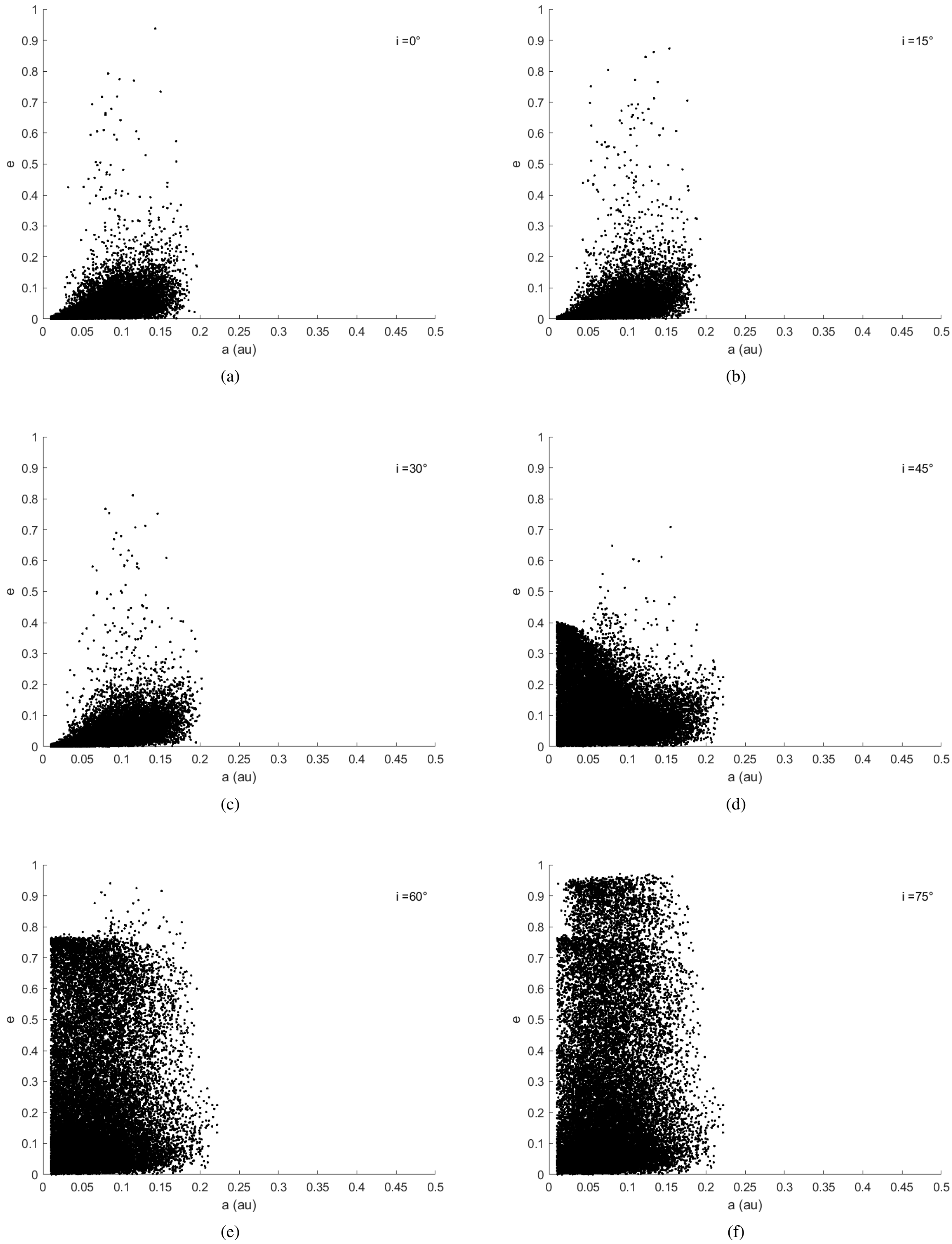


Fig. 3. Set of diagrams: the semimajor axis versus the eccentricity of the stable particles (in black) for the prograde case. In the different plots the ring particle inclination is: (*a*) 0°; (*b*) 15°; (*c*) 30°; (*d*) 45°; (*e*) 60°; and (*f*) 75°.

varies largely depending on the inclination, going from less than 0.05 au for the smallest scenario up to almost 0.4 au.

When the inclination is 90°, all particles are unstable, regardless of the mass of the planet. The figure also shows that the range of the ring size increases for larger inclinations. For prograde rings ($i < 90°$) the maximum extension is around 0.22 au, while for retrograde rings, where the systems are more stable, the ring reaches 0.38 au in radius.

The numerical results presented in Figs. 3 and 4 (prograde and retrograde cases, respectively) are the diagrams of the par-

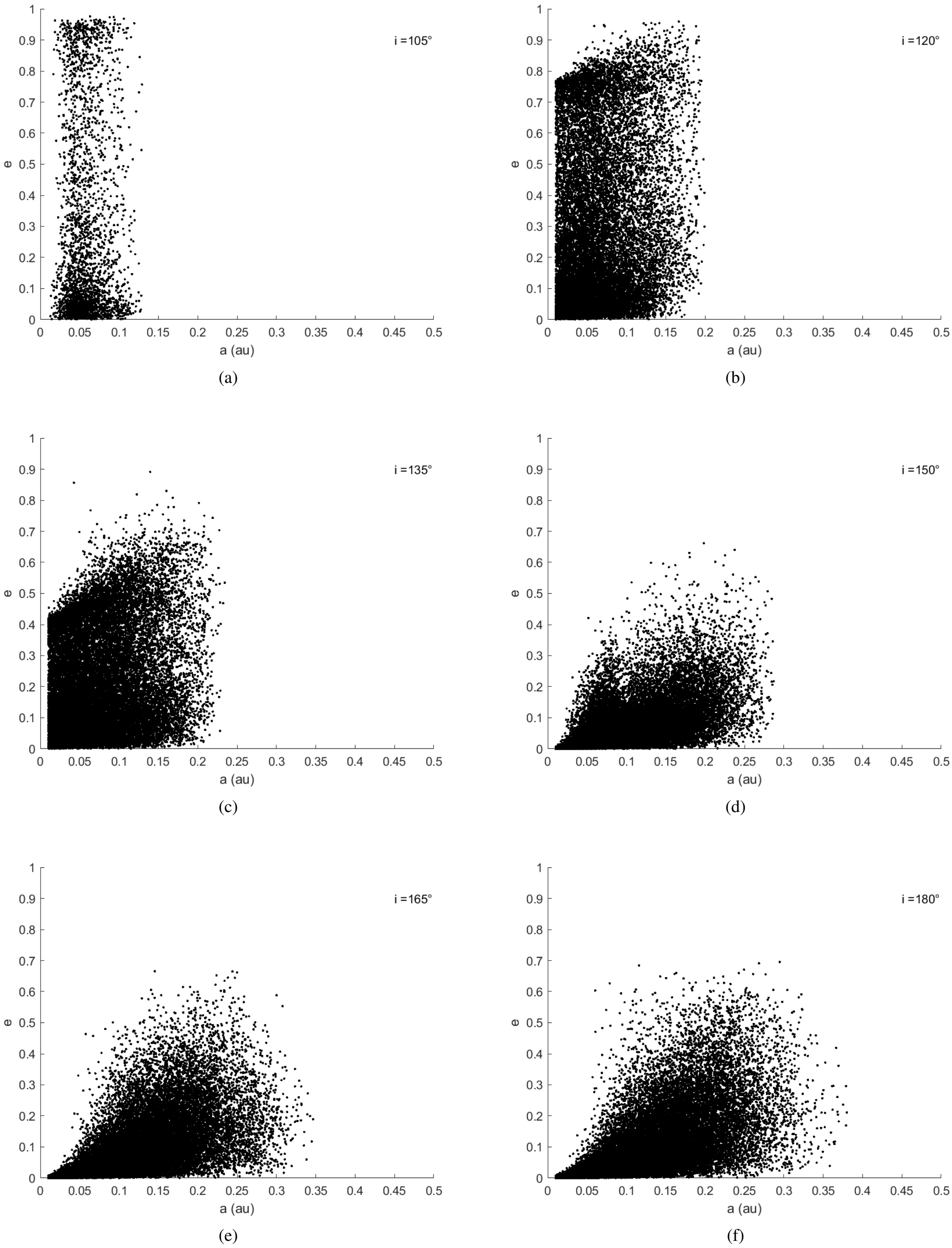


Fig. 4. Set of diagrams: the semimajor axis (a) versus the eccentricity of the stable particles (in black) for the retrograde case. The ring particle inclination is: (a) 105°; (b) 120°; (c) 135°; (d) 150°; (e) 165°; and (f) 180°. The diagram for $i = 90°$ is missing because no stable region exists in this case.

ticles' semimajor axis (a) versus particle eccentricity (e) for the stable particles. Each plot represents systems with different planet masses but the same ring inclination.

The initially circular ring particles quickly became eccentric, and this effect is more prominent for highly inclined rings. Figure 3 shows that for $i \leq 30°$ (panels a to c) the majority of particles are found within $e < 0.3$ and $a < 0.2$ au. For $i = 45°$ the stable particles are mostly confined to $e \leq 0.4$ and $a \leq 0.2$ au. In the range of $i = [60°, 75°]$, a significant number of particles present large eccentricities, reaching values close to unity with $a \leq 0.22$ au.

No stable particle is found with an inclination around 90°. In the retrograde case (Fig. 4), for $i = 105°$ the maximum semimajor axis distance of the stable region is around 0.13 au, and these

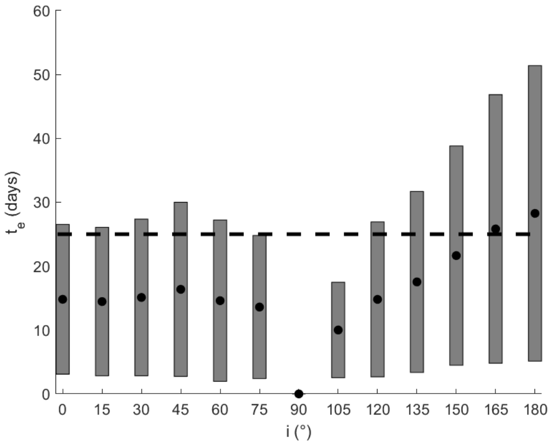


Fig. 5. Eclipse duration for different ring system inclinations. Each vertical bar represents the possible duration of the eclipses for a given inclination assuming the different ring extensions. The dot indicates the average value of each box, and the dashed line represents the observed eclipse (25 days).

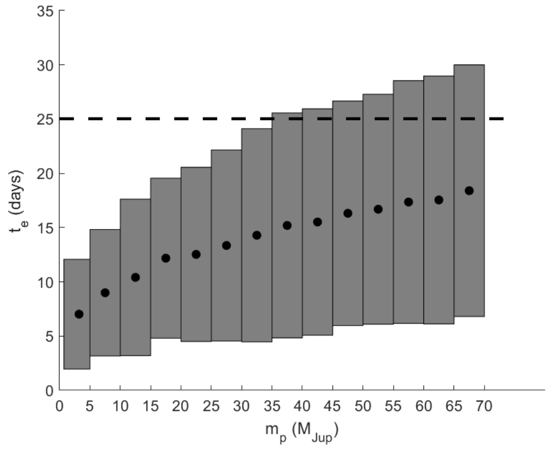


Fig. 6. Influence of the mass of the planet on the eclipse time, for prograde systems. The rectangles indicate the duration of the possible eclipses within a specific mass range, grouped every $\Delta m_p \approx 5\ M_{Jup}$.

values increase when the ring inclination is 180°, reaching as high as $a \sim 0.4$ au. The maximum particle eccentricity decreases as the ring inclination rises: The value of e goes from almost 1 for $i = 105°$ (Fig. 4a) to $e \leq 0.7$ for $i = 180°$ (Fig. 4f).

5.1. Eclipse duration

For each group of simulations, we calculated the theoretic eclipse time (t_e) according to Eq. (2), always assuming a face-on configuration. To identify the physical and orbital parameters of the PDS110b system from the numerical results, we assumed the planet semimajor axis to be 2 au and its orbital period to be 808 days. We considered the ring radius to be the semimajor axis of the most distant particle and considered the planet velocity derived in two different positions of its orbit, at pericenter and apocenter (extreme cases).

The interval of all possible theoretical eclipse times (t_e) for each group of simulations is presented in Figs. 5–11. In each plot the horizontal line corresponds to the observed eclipse duration (25 days), thus indicating which configurations (planet mass and eccentricity, ring extension, and inclination) for the PDS110b ring system could reproduce the observed events.

Figure 5 shows that both prograde and retrograde ring systems may explain the transit events when considering orbital inclinations smaller than 60° or higher than 120°. In the range $i = [75°–105°]$ the stable ring is not large enough to reproduce an eclipse that matches the observation. We now divide the discussion into two sections, the prograde and retrograde cases.

5.2. Prograde case

Out of our set of 1820 groups of simulations, half correspond to prograde orbits of the planet. In this case, the ring inclination is required to be smaller than 60° to produce an eclipse that lasts 25 days. It is most likely that in these systems the mass of the secondary companion is larger than 35 M_{Jup} (Fig. 6), which is consistent with the results presented in Fig. 3.

It is also possible to group the ring extension for the simulated systems according to the eccentricity of the planet (Fig. 7). Assuming the eclipse occurred at the apocenter of the planet, any eccentricity $e_p < 0.3$ is possible. However, if we consider that the eclipse took place at the pericenter, only a planet with a circular or slightly eccentric orbit ($e_p < 0.05$) could have a disk capable of consistently explaining the transit time, suggesting that a planet with a lower eccentricity is more plausible.

Planets with a higher eccentricity present a less extensive ring system, and this directly influences the eclipse duration. This is expected since the planet is reasonably close to the star (2 au) and since the outermost particles are increasingly perturbed by the star as the orbit becomes more eccentric.

We computed the eclipse duration for each of the 910 prograde cases, assuming that the planet is either in the pericenter or the apocenter (Fig. 8). Given that the velocity of the planet is maximum at the pericenter, the planet requires a ring with an extension of ~0.2 au to produce a 25-day eclipse when viewed face-on. Conversely, at the apocenter (thus, at a lower velocity), a ring extension from 0.1 to 0.2 au matches the observation. This range corresponds to less than 0.5 r_{Hill}, which is compatible with previous works.

5.3. Retrograde case

Retrograde systems can also sustain a ring that is large enough to be consistent with a transit that lasts 25 days. In fact, given the more extensive distribution of stable particles (see Fig. 4), more retrograde systems fit the eclipse than prograde orbit systems. When grouping the simulations according to the mass of the planet (Fig. 4), we see that only systems where $m_p < 5\ M_{Jup}$ cannot explain the eclipse.

Regarding the possible eccentricities of the planet, if the eclipse happens at the apocenter (right panel in Fig. 10), any value smaller than 0.5 is allowed. However, at the pericenter, where the orbital velocity is at its maximum, only systems with a smaller eccentricity ($e_p < 0.25$) are plausible solutions.

Figure 11 shows the eclipse time as a function of the ring extension of each retrograde system. In this case, the eclipse could be explained by some systems with a ring radius from 0.1 au up to 0.25 au, depending on whether we assume the eclipse happens at the pericenter or apocenter of the planetary orbit. Due to the stability of the retrograde system, the size of the ring could be much larger than what is necessary to explain the eclipse, reaching almost 0.4 au (~0.95 r_{Hill}) in extreme cases.

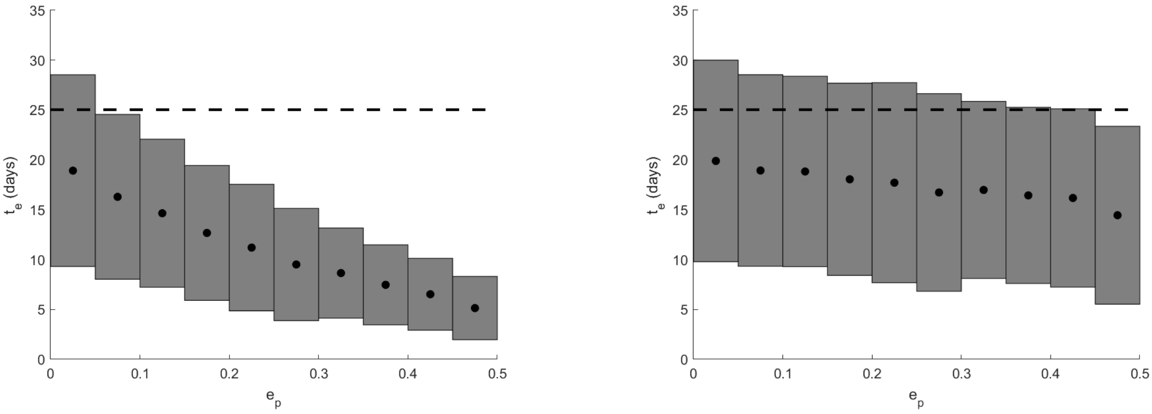


Fig. 7. Time span of the duration of the eclipses for a prograde ring grouped according to the eccentricity of the planet (e_p). *Panel on the left*: the eclipse is occurring at the pericenter. *Right*: the eclipse is occurring during the planet passage at the apocenter. The dashed line represents the observable eclipse (25 days).

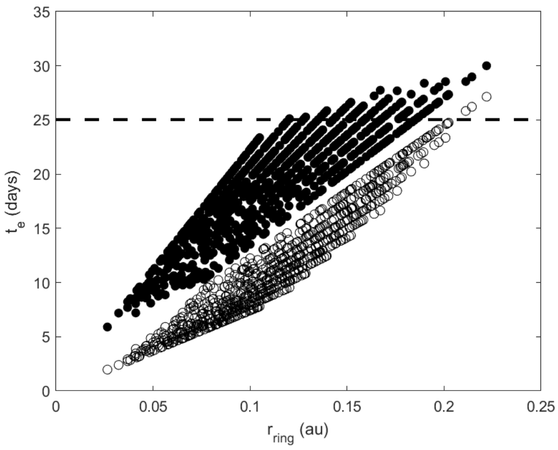


Fig. 8. Eclipse time according to the ring radius for planets with a prograde orbit. The eclipse duration changes if it is assumed that the event occurs at the pericenter of the orbit (open circles) or the apocenter (filled circles). The dashed line indicates the observed eclipse.

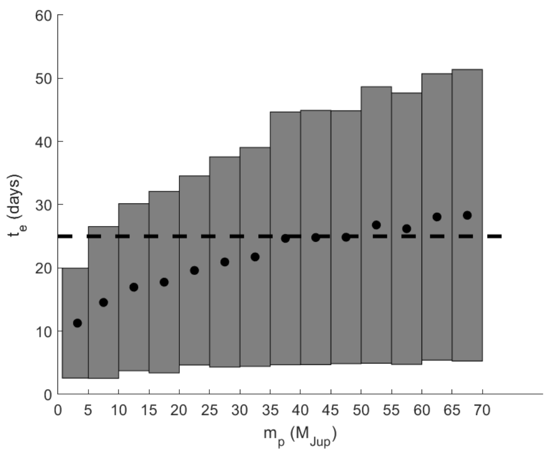


Fig. 9. Influence of the mass of the planet on the eclipse time, for retrograde systems. The rectangles indicate the duration of the possible eclipses within a specific mass range, grouped every $\Delta m_p \approx 5\,M_{\rm Jup}$.

6. Final comments

The two eclipses of the star PDS110 in 2008 and 2011 were initially interpreted as results of the transit of the edge of a gigantic and inclined ring (Osborn & Rodriguez 2017). This ring could be around an unseen secondary companion, either a brown dwarf or a giant planet (named PDS110b). This secondary would be 2 au from the star, with a mass $>1.8\,M_{\rm Jup}$. No other constraints on the system could be made through the observational data.

We proposed to find possible limits on the mass and eccentricity for the secondary body, as well as the ring radius and inclination that could produce an equivalent transit duration assuming we have a face-on view of the system. To do so, we performed numerical simulations, assigning random values for the parameters of the system within the ranges derived from the observations.

Initially, we found that the limit of the stability zone for a circumplanetary disk sets the maximum eccentricity of the planet as 0.5, which is in agreement with an analytical approach. If the planet is more eccentric, the ring is more disturbed by the star, and its size is not large enough to create a 25-day eclipse.

We also simulated rings with inclinations from 0° to 180°. The largest prograde stable ring system we found filled a fraction of $\sim 0.5\,r_{\rm Hill}$, compatible with theoretical studies. In retrograde simulations, this stability zone can extend to almost one Hill radius ($\sim 0.95\,r_{\rm Hill}$).

The approach we followed to determine the stability resulted in a large ring and cannot be directly compared to the method presented by Speedie & Zanazzi (2020). Our instability criteria rely on ejection or collision, while their analysis is related to a torque balance between the tidal torques from the star and a planet, and they also have a stricter stability classification related to the variation in the inclination and eccentricity of the particle. Furthermore, the torque from the planet is directly dependent on the J_2 coefficient, which is an unknown parameter for the PDS110b and would introduce an extra degree of freedom in our analysis.

The eclipse duration depends on the orbital velocity of the planet. Thus, we computed the extreme cases for transit at the pericenter and at the apocenter, both for prograde and retrograde ring systems. When the ring inclination is smaller than 90°, it requires a more massive planet ($m_p > 35\,M_{\rm Jup}$) and its eccentricity is limited to 0.2 for transit at the pericenter. If the eclipse

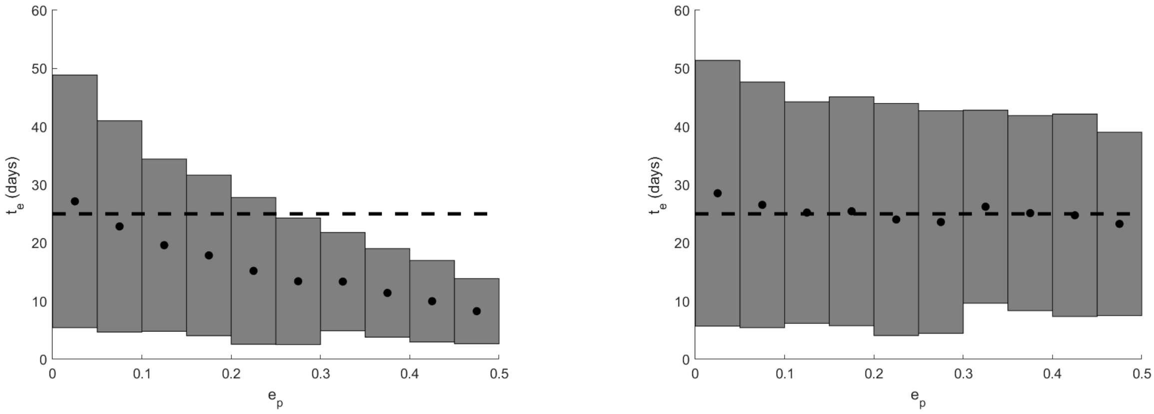


Fig. 10. Time span of the duration of the eclipses for a retrograde system according to the eccentricity of the planet (e_p). *Panel on the left*: the eclipse is occurring at the pericenter. *Right*: during the planet passage at the apocenter.

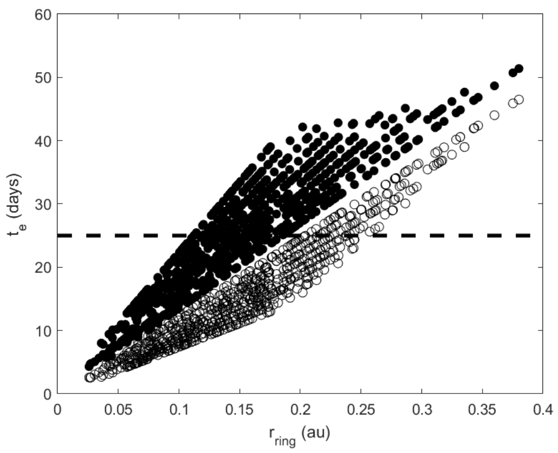


Fig. 11. Eclipse time according to the ring radius for planets with a retrograde orbit. The eclipse duration changes if it is assumed that the event occurs at the pericenter of the orbit (open circles) or the apocenter (filled circles). The dashed line indicates the observed eclipse.

occurs at the apocenter, any value of $e_p < 0.5$ could explain the eclipse. In a retrograde configuration, the imposed constraints are looser since the eclipse could be explained for systems with $m_p > 5\,M_{Jup}$.

Although the retrograde ring system may have a broader range of possible configurations to explain the eclipse, we consider a prograde system to be more likely since it is hard to justify the origin of such a large ring (~0.2 au) in retrograde motion. In our Solar System, we do not have any example of a retrograde ring system, just some irregular moons of Jupiter, Saturn, and Neptune; in these cases, the moons were captured into orbit by the planet's gravity.

In summary, the most likely result from our simulations is the configuration for the PDS110b system with a ring radius in the range [0.1–0.2] au and an inclination $i < 60°$. The planet probably has a small eccentricity ($e < 0.05$) and is probably very massive $m_p > 35\,M_{Jup}$.

It is worth recalling that we assumed that we are always observing the system face-on in our analysis. If the disk is inclined in the sky plane, the observed radial extension of the ring is reduced, and therefore many of the possible configurations that we considered acceptable vanish.

Since the results we present are derived from numerical simulations carried out for more than 6000 yr (10^4 orbital periods), it is beyond the scope our analysis to explain why the predicted dimming event in 2017 was negative after two positive detections in 2008 and 2011. Osborn & Kenworthy (2019) argued that these eclipses might be some aperiodic event and not related to the transit of an exoring.

Acknowledgements. The authors are grateful to the anonymous referee that contributed significantly to the improvement of the paper. This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and Fundação de Amparo a Pesquisa no Estado de São Paulo (FAPESP) Proc. 2016/24561-0.

References

Aizawa, M., Uehara, S., Masuda, K., Kawahara, H., & Suto, Y. 2017, AJ, 153, 193
Aizawa, M., Masuda, K., Kawahara, H., & Suto, Y. 2018, AJ, 155, 206
Akinsanmi, B., Santos, N., Faria, J., et al. 2020, A&A, 635, L8
Barnes, J. W., & Fortney, J. J. 2004, AJ, 616, 1193
Brown, A. G., Vallenari, A., Prusti, T., et al. 2016, A&A, 595, A2
Domingos, R. C., Winter, O. C., & Yokoyama, T. 2006, MNRAS, 373, 1227
Hamilton, D. P., & Burns, J. A. 1992, Icarus, 96, 43
Heising, M., Marcy, G., & Schlichting, H. 2015, ApJ, 814, 81
Hunter, R. 1967, MNRAS, 136, 245
Kenworthy, M. A., & Mamajek, E. E. 2015, ApJ, 800, 126
Mamajek, E. E., Quillen, A. C., Pecaut, M. J., et al. 2012, AJ, 143, 72
Mayor, M., & Queloz, D. 1995, Nature, 378, 355
Osborn, H. P., & Kenworthy, et al. 2019, MNRAS, 485, 1614
Osborn, H. P., & Rodriguez, et al. 2017, MNRAS, 471, 740
Rein, H., & Spiegel, D. S. 2014, MNRAS, 446, 1424
Rieder, S., & Kenworthy, M. A. 2016, A&A, 596, A9
Schlichting, H. E., & Chang, P. 2011, AJ, 734, 117
Speedie, J., & Zanazzi, J. 2020, MNRAS, 497, 1870
Wolszczan, A., & Frail, D. A. 1992, Nature, 355, 145

APPENDIX B – Machine Learning approach to map stability around planets

Copy of the published version of the paper 'Machine learning approach for mapping stability regions around planetary systems' in the journal Astronomy & Astrophysics, 693.

A&A, 693, A246 (2025)
https://doi.org/10.1051/0004-6361/202451831

Astronomy & Astrophysics

Machine-learning approach for mapping stable orbits around planets

Tiago F. L. L. Pinheiro[1,⋆], Rafael Sfair[1,2,3], and Giovana Ramon[1]

[1] Grupo de Dinâmica Orbital e Planetologia, São Paulo State University, UNESP, Guaratinguetá, CEP 12516-410, São Paulo, Brazil
[2] Eberhard Karls Universität Tübingen, Auf der Morgenstelle, 10, 72076 Tübingen, Germany
[3] LESIA, Observatoire de Paris, Université PSL, CNRS, Sorbonne Université, 5 place Jules Janssen, 92190 Meudon, France

Received 7 August 2024 / Accepted 3 December 2024

ABSTRACT

Context. Numerical N-body simulations are typically employed to map stability regions around exoplanets. This provides insights into the potential presence of satellites and ring systems.
Aims. We used machine-learning (ML) techniques to generate predictive maps of stable regions surrounding a hypothetical planet. This approach can also be applied to planet-satellite systems, planetary ring systems, and other similar systems.
Methods. From a set of 10^5 numerical simulations, each incorporating nine orbital features for the planet and test particle, we created a comprehensive dataset of three-body problem outcomes (star-planet-test particle). Simulations were classified as stable or unstable based on the stability criterion that a particle must remain stable over a time span of 10^4 orbital periods of the planet. Various ML algorithms were compared and fine-tuned through hyperparameter optimization to identify the most effective predictive model. All tree-based algorithms demonstrated a comparable accuracy performance.
Results. The optimal model employs the extreme gradient boosting algorithm and achieved an accuracy of 98.48%, with 94% recall and precision for stable particles and 99% for unstable particles.
Conclusions. ML algorithms significantly reduce the computational time in three-body simulations. They are approximately 10^5 times faster than traditional numerical simulations. Based on the saved training models, predictions of entire stability maps are made in less than a second, while an equivalent numerical simulation can take up to a few days. Our ML model results will be accessible through a forthcoming public web interface, which will facilitate a broader scientific application.

Key words. methods: numerical – planets and satellites: dynamical evolution and stability

1. Introduction

The population of known exoplanets has grown every year and has prompted extensive research of the stability of extrasolar planetary systems. To investigate the potential existence of satellites or ring systems around these exoplanets, numerical N-body simulations are frequently employed. These simulations map the stability regions around a planet based on a grid of orbital parameters as initial conditions for the N-body integration.

Several studies expanded this analysis to various contexts in different gravitational settings. Hunter (1967) examined the stability of satellites orbiting Jupiter under the influence of solar gravity. Their findings indicate that the stable zone for prograde motion extends to 0.45 of a Hill radius (r_{Hill}), while for retrograde motion, it extends to 0.75 r_{Hill}. Neto & Winter (2001) investigated the gravitational capture in the Sun-Uranus-satellite system. They employed a diagram of initial conditions of semimajor axis versus eccentricity to identify stable regions around Uranus, considering particles with various initial inclinations and arguments of pericenter.

Holman & Wiegert (1999) numerically investigated the stability of S-type and P-type planet orbits around a binary star system and treated the planet as a test particle. Their work resulted in an empirical expression for the critical semimajor axis of a stable particle that initially is on a circular and planar orbit, relating it to binary distance, mass ratio, and eccentricity. The authors analyzed a range of binary mass ratios $\mu = [0–0.9]$ and eccentricities e $= [0–0.8]$.

Domingos et al. (2006) examined the stability of hypothetical satellites orbiting extrasolar planets with a mass ratio $\mu = 10^{-3}$. They derived an empirical expression for the stability boundary in prograde ($i = 0°$) and retrograde ($i = 180°$) orbits, formulated in terms of the Hill radius and the eccentricities of the planet and the satellite.

Rieder & Kenworthy (2016) conducted N-body simulations to analyze the stability of a potential ring system around the candidate planet J1407b, thereby imposing constraints on the orbital and physical parameters of this system. For the PDS110b system, Pinheiro & Sfair (2021) performed millions of numerical simulations of the three-body problem with the aim to refine parameters that remained undefined based on observational data, including the mass and eccentricity of an unseen secondary companion, as well as the ring inclination and radial size.

While these studies provided valuable insights, conducting numerical simulations like this for many systems incurs substantial computational expense. To address this challenge, a novel method to expedite this process involves the use of Machine-Learning (ML) algorithms. In this approach, N-body simulations generate a dataset for training ML models, which can then predict outcomes for unseen data more efficiently.

The identification of stability regions using ML methods was demonstrated in previous studies. Tamayo et al. (2016) applied ML algorithms to predict the stability of tightly packed planetary

⋆ Corresponding author; francisco.pinheiro@unesp.br

systems that consist of three planets orbiting a central star. Their models achieved a predictive accuracy of 90% over the test set, and they were faster by three orders of magnitude than direct N-body simulations.

Tamayo et al. (2020) developed the code called stability of planetary orbital configurations classifier (SPOCK). This ML model is capable of predicting the stability of compact systems with three or more planets. This approach is 10^5 times faster than numerical simulations.

Cranmer et al. (2021) introduced a probabilistic ML model that predicts the stability of compact multiplanet systems with three or more planets, including when these systems are likely to become unstable. This model demonstrates an accuracy higher by two orders of magnitude in predicting instability times than analytical estimators.

Expanding on this approach, Lam & Kipping (2018) employed deep neural networks to predict the stability of initially coplanar circular P-type orbits for circumbinary planets. Their method achieved an accuracy of at least 86% for the test set, which further demonstrates the potential of ML in analyzing orbital stability.

Building upon the successes of previous ML applications in orbital stability analysis, we propose using this approach to predict a stable-region map surrounding a single-planet system. This method can also be extended to analyze the stability of other analogous systems, such as planet-satellite pairs, planetary rings, and binary minor planets, provided they lie within the trained parameter space of the ML model.

We conducted a series of dimensionless numerical simulations involving an elliptical three-body problem (star, planet, and a particle orbiting the planet). Each simulation represents a distinct initial condition within our dataset, which is used to train and evaluate ML algorithms. Expanding on the work of Domingos et al. (2006), our study explores a wide range of orbital and physical parameters, including the mass ratio of the system and several orbital elements of the particle and planet.

This comprehensive approach would be computationally demanding if it were solved purely through numerical simulations, especially when the number of grid parameters to be analyzed is increased. Running a single three-body problem may be relatively fast, but a broader set of initial conditions may significantly increase computational costs. This highlights the advantages of our ML method.

This paper is structured as follows: Section 2 introduces key concepts and methods for our analysis and introduces the ML algorithms. Section 3 describes our numerical model and the dataset preparation. Section 4 describes our treatment of imbalanced classes. Section 5 evaluates the performance of our ML algorithms. Section 6 compares the predictions made by our best-performing ML model with the results from Domingos et al. (2006) and Pinheiro & Sfair (2021), and with a stability map of the Saturn satellites. Finally, Section 7 presents our concluding remarks and implications of this study.

2. Supervised learning

This section presents the fundamental concepts and techniques for our study. These definitions and approaches are referenced and compared throughout the subsequent sections, particularly in our results and discussion.

Supervised learning includes the desired solution (labels) in the training data. These labels can be composed of a continuous numerical value (regression task) or of discrete values that represent different classes (classification task).

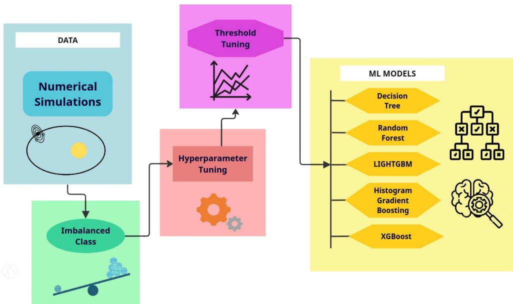


Fig. 1. Flowchart illustrating the workflow.

Classification tasks are techniques that use training samples (or instances) during the learning process to develop a generalization of the classes and predict given unknown new instances. We address binary classifications of stable and unstable particles to predict stability maps within a specific planetary system.

The dataset was split into three subsets: training, validation, and test data. The training data were used by the algorithm to find the best generalization features for different classes. The validation data provided an unbiased evaluation of the model effectiveness, and they were used to tune the model hyperparameters and thresholds, select the best ML algorithm, and detect under- and overfitting (Shalev-Shwartz & Ben-David 2014).

The cross-validation technique involves partitioning the training data into subsets. Specifically, we split the training data into five subsets. Subsequently, we trained and evaluated the model on distinct subsets, which was followed by averaging the outcomes to estimate the unbiased error on the training set of the model (Fushiki 2011).

After training several models with different configurations multiple times on the reduced training set and evaluating them through a cross-validation, we identified the best-performing model. Subsequently, we retrained the selected model using the complete training dataset and evaluated the final performance of the ML model using holdout data (test dataset).

We divided our work into five main steps, as shown in Figure 1, which sketches the workflow of our method. The process began with the generation of a dataset from numerical simulations. We then addressed the issue of imbalanced classes through resampling techniques. Using a cross-validation, we tuned the hyperparameter and explored optimal threshold values to establish the best-configuration model. The final step involved assessing the performance of the best model.

2.1. Hyperparameter tuning

Hyperparameters are specific algorithm parameters that must be set prior to the training process, and the model performance directly depends on them (Weerts et al. 2020). The search for the best hyperparameters was carried out in two different ways. The first way was a grid search, and the second way was a random search. Both ways were validated using a cross-validation.

The grid search systematically explored all possible combinations of the hyperparameters. In contrast, the random search randomly selected some combinations that were evaluated, which is useful for an extensive number of potential combinations.

2.2. Threshold tuning

In binary classification tasks, classifier algorithms in general employ a default threshold value of 0.5. This means that when

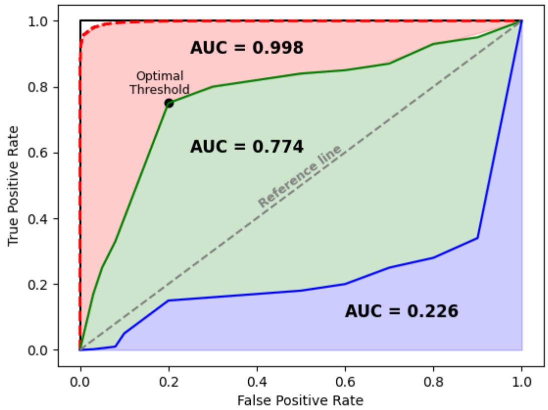


Fig. 2. ROC curve representation of three hypothetical models. The black curve behaves like an ideal model, and the green and blue curves represent good and poor classifiers, respectively. The dashed gray line is the reference line. The red curve represents the performance of our best-performing ML model.

the probability that an instance is in a positive class is ≥0.5, it is assigned to the positive class; otherwise, it is classified as negative. However, for an imbalanced dataset, it is generally inappropriate to use 0.5 as the threshold values (Zou et al. 2016).

The receiver operating characteristic (ROC) curve is a frequently employed tool for determining the optimal threshold value (Géron 2022). The ROC curve is a probability curve that plots the true-positive rate (hereafter referred to as recall) versus the false-positive rate for different threshold probabilities from 0 to 1. Figure 2 provides an illustrative example of ROC curves for three hypothetical models with varying performance levels, and also the ROC curve of our best performance ML model (see Sect. 5).

The true-positive rate is the ratio of positive instances that are correctly predicted, while the false-positive rate represents the fraction of negative instances that are incorrectly predicted as positive. The ROC curve always begins at point (0,0), where the threshold is 1, signifying that all instances are predicted as negative. It ends at (1,1), where the threshold is 0, indicating that all instances are predicted as positive.

The area under the curve (AUC) represents the area under the ROC curve. It provides a measure of the model ability to differentiate classes. In a perfect model (exemplified by the black curve in Figure 2), the positive class is always predicted with an accuracy of 100%, regardless of the threshold parameter. This results in an AUC of 1.

Good predictors tend to have an AUC value close to 1. For instance, the green curve in Figure 2 has an AUC of 0.77, indicating that the model has a probability of 77% of successfully distinguishing between positive and negative classes. The dashed red line represents the ROC curve of our best-performing ML model with an AUC of 0.998. This model is discussed in Section 5.

The reference line (dashed gray line) represents an AUC of 0.5, signifying that the model cannot differentiate between classes. This results in random guesses for predictions. Conversely, the blue curve represents a poorly performing model with an AUC of 0.23, where the majority of positive instances are predicted as negative.

The optimal threshold (the black dot in Figure 2) can be determined as the point closest to the upper left corner of the ROC curve (0,1), or the point that maximizes the distance from the reference line where the true-positive rate is high and the false-positive rate is low.

The concepts and methods outlined in this section provide the framework for our subsequent analysis. We refer to these definitions and compare their effectiveness below, in particular when we evaluate our results and discuss their implications.

2.3. Classification algorithms

We tested eight distinct ML algorithms to determine the optimal approach: nearest neighbors, naive Bayes, artificial neural network, and the tree-based methods decision tree, random forest, XGBoost, light gradient boosting machine (LightGBM), and histogram gradient boosting. Finally, we ran a genetic algorithm to verify our findings.

Because all tree-based algorithms demonstrated better and comparable performance, we focused our analysis on these methods. With the exception of decision tree, all other tree-based classifiers are ensemble methods that use a group of weaker learners (in this case, an ensemble of decision trees) to enhance their predictions. The collective perspective of a group of predictors yields better predictions than a single predictor (Géron 2022). We give further details of the fundamental concepts of each of these tree-based algorithms in Appendix A.

3. Dataset preparation

In this section, we detail the method with which we investigated the orbital stability of a three-body problem (star, planet, and particle) through an ensemble of dimensionless numerical simulations. Each simulation represents a distinct initial condition for a hypothetical planetary system.

We began by defining the mass parameter μ as

$$\mu = \frac{m_{\rm p}}{M_{\rm s} + m_{\rm p}}, \tag{1}$$

where $m_{\rm p}$ and $M_{\rm s}$ are the masses of the planet and star, respectively.

The survey including 4150 confirmed exoplanets from Schneider et al. (2011), reveals that 95.71% of them have an equivalent $\mu \le 10^{-2}$, and 2.05% of these have $\mu < 10^{-5}$. Thus, we restricted our analysis to this range of mass parameters, and within this, we adopted a random uniform distribution.

Throughout all numerical simulations, the star and planet masses were set by $M_{\rm s} = 1 - \mu$ and $m_p = \mu$, respectively. The planet semimajor axis was set to 1, and its collision radius ($r_{\rm c}$) was defined as 5% of the Hill radius ($R_{\rm Hill}$) computed at the pericenter,

$$r_c = 0.05(1 - e_p)\sqrt[3]{\frac{\mu}{3}}. \tag{2}$$

This choice for r_c means that 55.6% of the planets in the catalog of Schneider et al. (2011) have a collision radius that is smaller than 5% of their Hill radii, as shown in the histogram in Figure 3.

The planet eccentricity ($e_{\rm p}$) and true anomaly ($f_{\rm p}$) were randomly uniformly sampled within the specified ranges of e_p = [0–0.99] and f_p = [0°–360°]. The remaining orbital elements for the planet inclination ($i_{\rm p}$), arguments of pericenter ($\varpi_{\rm p}$), and longitude of the node ($\Omega_{\rm p}$) were set to zero.

The test particle was modeled orbiting a planet and gravitationally disturbed by the star. We adopted a random uniform

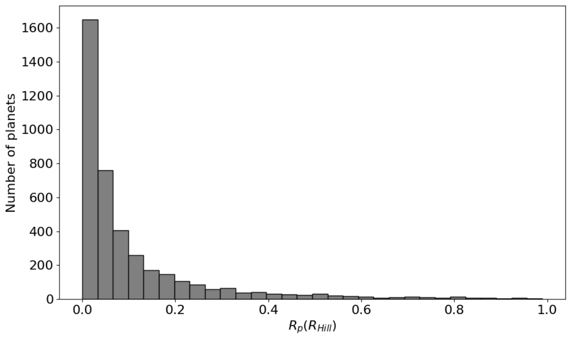


Fig. 3. Histogram of the number of confirmed planets in units of their Hill radius from a survey consisting of 4150 different planets (Schneider et al. 2011).

distribution to select the particle orbital initial conditions for the semimajor axis from 1.1 r_c to 1 $R_{\rm Hill}$, an eccentricity from 0 to 0.99, an inclination between 0° and 180°, and arguments of pericenter, longitude of the node, and true anomaly ranging from 0° to 360°.

We numerically integrated using the REBOUND package and the IAS15 integrator (Rein & Spiegel 2015) for a time span of 10^4 orbital periods of the planet. We recorded every time when the particle collided with the planet or was ejected from the system, which occurred when the particle reached a hyperbolic orbit (eccentricity $e \geq 1$) or when its semimajor axis was $a > 1$.

In total, we ran 10^5 numerical systems, and the overall outcome was as follows: 11.83% of the particles remained stable throughout the entire integration, while 88.17% were unstable. Of the unstable particles, 53.56% collided with the planet, and 46.44% were ejected from the system.

The outcomes of the numerical simulations were used to build a dataset for training ML algorithms. This dataset consisted of nine features that were the initial conditions of each numerical simulation: the mass ratio of the system, semimajor axis, inclination, argument of pericenter, and longitude of the node of the particle, and the eccentricity and true anomaly of the planet and the particle. We labeled each initial condition with the numerical results as either a stable or an unstable system.

Figure 4 shows four histograms of the number of numerical simulations in relation to the particle semimajor axis (upper left panel), the eccentricity (upper right panel), the inclination (lower left panel), and the planet eccentricity (lower right panel).

As the eccentricity of the planet increases, the number of stable systems is slightly reduced. Nevertheless, we can find stable particles even for highly eccentric planets. The particle eccentricity has a more distinct effect on the number of stable systems: no stable orbits are found for $e > 0.8$. This is expected since particle orbits experience stronger gravitational disturbances at their pericenters, which increases the quantity of particles that collide.

The number of stable systems is also strongly influenced by the inclination of particles relative to the orbital plane of the planet. Our set contains no stable systems for inclinations between 80° and 110°, and retrograde systems tend to have more stable orbits than prograde ones.

A significant number of collisions occurs closer to the planet, and the number of stable systems increases with the initial particle distance, reaching a peak at 0.3 $R_{\rm Hill}$. After this peak, the quantity of stable systems declines, and only a few systems remain stable beyond 0.7 $R_{\rm Hill}$; most particles are ejected.

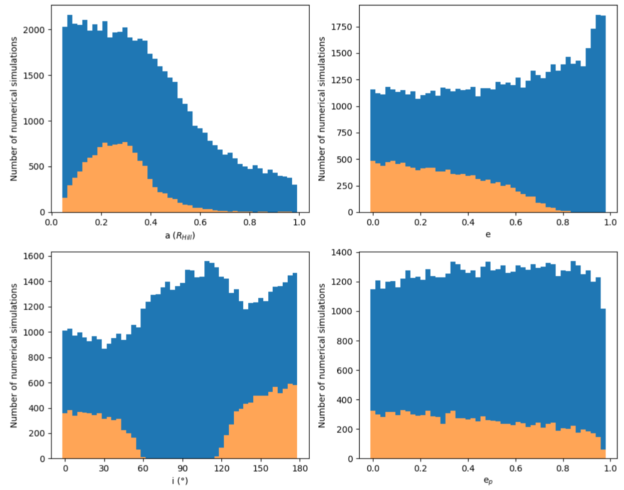


Fig. 4. Histogram of the distribution of stable (orange) and unstable (blue) systems across the range of particle semimajor axes (upper left panel), particle eccentricities (upper right panel), particle inclinations (lower left panel), and planet eccentricities (lower right panel).

Other features, such as the mass ratio of the system, the argument of pericenter, the longitude of the ascending node, and the true anomaly, were distributed regularly across the range of their parameters.

4. Imbalanced class

The numerical results demonstrate that the classes are disproportionate in the dataset by a factor 1:10, with the unstable class being much larger than the stable class. This imbalance could lead to biases in the classification and might impair the effectiveness of the various classifier algorithms when they try to generalize (Zheng et al. 2015). ML algorithms typically assume that classes are evenly distributed, which can causes the minority class to be classified poorly (Kumar & Sheshadri 2012; Carruba et al. 2023).

Resampling the training data is one strategy that can be used to solve this problem. One option is undersampling. This technique removes instances from the majority class. The weakness of this method is that it may lose information by removing part of the data (Mohammed et al. 2020). Another resampling process is oversampling, which involves increasing the size of the minority class until the classes are balanced. We tested the performance of four different types of oversampling: random oversampling, the synthetic minority oversampling technique (SMOTE), borderline SMOTE, and adaptive synthetic sampling (ADASYN). In Appendix B, we provide additional details on the key concepts of each of these resampling techniques.

According to Carruba et al. (2023), a severely imbalanced dataset can occur when the class ratio is 1:100, and in this scenario, resampling methods may be necessary. After testing the performance of our algorithms with various resampling techniques, we observed a marginal improvement, except for the LightGBM, which increased by 13% in identifying stable particles. This was achieved solely through the resampling technique, and no hyperparameters or thresholds were tuned. With this improvement, the LightGBM performance is almost as good as the best-performing model without any resampling.

5. Performance evaluation

Evaluation metrics enable the assessment of the classifier performance. The confusion matrix is a statistical table that maps the prediction results by showing the quantity of correctly and incorrectly predicted data (Stehman 1997). While accuracy measures the rate at which a model correctly predicts, this metric may not be reliable for imbalanced datasets. A confusion matrix displays the actual number of instances for each class in its rows and the predicted number of instances for each class in its columns. It is divided into four categories: true positive (TP) and true negative (TN), where the algorithm correctly predicts positive and negative classes; and false positive (FP) and false negative (FN), where the algorithm incorrectly predicts positive and negative classes.

These four categories are used to measure recall, specificity, precision, F1-score, and accuracy. The accuracy is the ratio of instances that are correctly predicted by the algorithm, given by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{3}$$

The specificity or true negative rate is the ratio of correctly predicted instances within the actual negative class,

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{4}$$

and the false-positive rate is given by 1 − Specificity.

The recall or the true-positive rate is the ratio of correctly classified positive instances to the total number of actual positive instances,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5}$$

The precision is the ratio of actual positive instances within the predicted positive class,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{6}$$

The F1-score combines precision and recall by calculating their harmonic mean,

$$\text{F1-score} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right). \tag{7}$$

While accuracy measures the rate at which a model correctly predicts, this metric may not be reliable for imbalanced datasets. The F1 scores measure the overall quality of a model, ranging from 0 to 1, where 1 signifies a model that perfectly classifies instances and 0 indicates a model that fails to classify any instance correctly.

5.1. Results and comparison of different algorithms

The ML outcomes were evaluated using a dataset generated from numerical simulations (see Sect. 3). Our binary approach involved classifying systems as either stable or unstable. The final dataset consisted of 100 000 simulations, divided into three subsets: 80% for training and validation data, and 20% for testing data. In the presence of imbalanced classes and to achieve better performance in classifier algorithms, we tested five distinct resampling methods. We used cross-validation techniques to identify the best hyperparameters, resampling method, and the optimal threshold value for each algorithm.

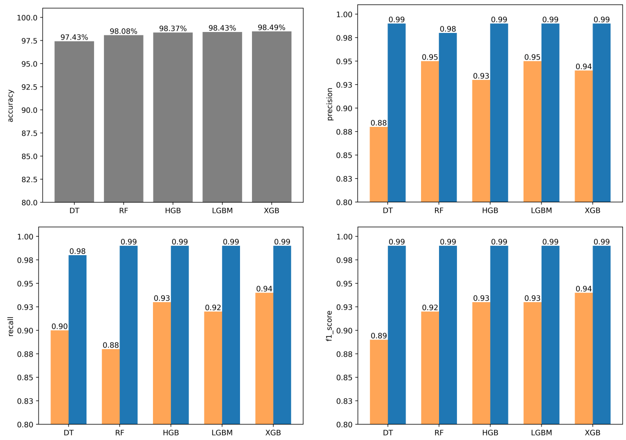


Fig. 5. Performance of the best model of the five tested algorithms. The upper panels show accuracy and precision, and the lower panels show recall and F1-score. The stable and unstable classes are represented by orange and blue bars, respectively. The algorithms are abbreviated as follows: DT, RF, HGB, LGBM, and XGB.

The best performance among the five algorithms was achieved by XGBoost, without using any resampling techniques, with a threshold value of 0.4, and setting some hyperparameters as follows:

I. booster = gbtree.Booster is a hyperparameter for the type of boosting model used during the training stage. Gbtree means gradient boosted trees.
II. eta = 0.2. Eta is the learning rate, and mathematically, it scales the contribution of each tree added to the model.
III. scale_pos_weight = 0.4. Controls the balance of positive and negative class weights.
IV. max_depth = 8. The maximum depth for each tree.

This result was also verified using the genetic algorithm. Genetic algorithms mimic the process of genetic evolution and are capable of selecting the most optimal algorithm and the best hyperparameters for a given dataset (Chen et al. 2004).

Figure 5 shows the performance of the five algorithms using four distinct metrics. The upper left panel displays the accuracy, and the precision for each class is presented in the upper right panel. The lower left panel shows the recall, and the lower right panel shows the F1-scores. Stable classes are represented by orange bars, and unstable classes are shown with blue bars.

All algorithms demonstrated a comparable accuracy performance. XGBoost achieved the highest accuracy at 98.48%, while Decision Tree exhibited the lowest score at 97.43%, showing a marginal difference of 1.05%.

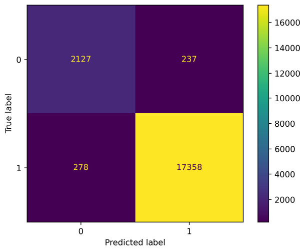
An important consideration here is that the accuracy measures how correctly the model classifies whether the system is stable or unstable across all predicted data. Therefore, this favorable result may be influenced by the proportion of imbalanced classes, with the quantity of unstable instances being approximately 7.5 times larger than stable ones in our dataset. The algorithms achieved a recall and precision of 99% for all classes, except for the precision of Random Forest and the recall of the Decision Tree algorithm, which were 98%.

In terms of stable class performance, LightGBM and Random Forest achieved the highest precision value. They correctly classified 95% of their predictions. They identified 92 and 88% of the actual stable instances (recall), respectively. XGBoost attained the same percentages for precision and recall, leading

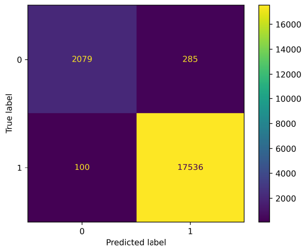
Pinheiro, T. F. L. L., et al.: A&A, 693, A246 (2025)

Table 1. Area under the ROC curve for the best model of each algorithm, sorted in descending order of performance.

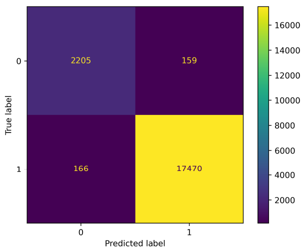
Model	AUC
XGBoost	0.9978
LightGBM	0.9975
Histogram Gradient Boosting	0.9974
Random Forest	0.9957
Decision Tree	0.9554



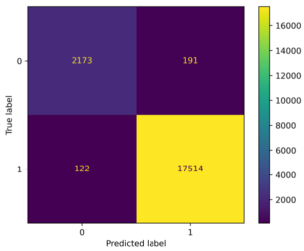
(a) Decision Tree



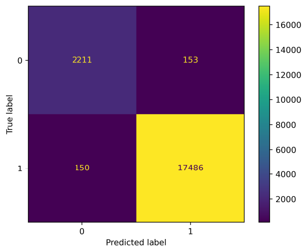
(b) Random Forest



(c) Histogram Gradient Boosting



(d) LightGBM



(e) XGBoost

Fig. 6. Confusion matrices for the best models of five ML algorithms.

to the highest F1-score of 94%, while LightGBM and Random Forest achieved 93 and 92%, respectively.

The area under the ROC curve (AUC) provides a measure of the overall model performance, with a value of 1 representing perfect classification (Figure 2). Table 1 compares the AUC values for each algorithm. XGBoost outperformed other models, closely followed by LightGBM and Histogram Gradient Boosting. The Decision Tree algorithm shows a slightly lower performance than the ensemble methods.

Figure 6 shows the confusion matrices for each algorithm and provides the statistical results from the testing dataset of 20 000 instances. Class 0 represents a stable system, and class 1 is unstable. All algorithms demonstrate a satisfactory performance. Even the lowest-performing model (Decision Tree) correctly predicts at least 19 485 instances.

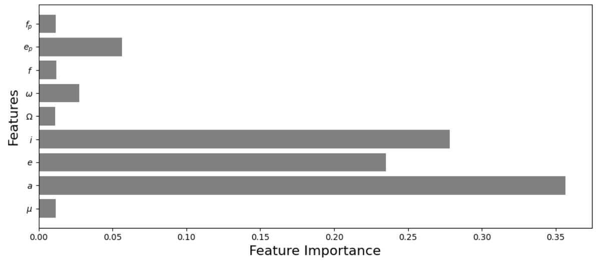


Fig. 7. Feature importance derived from the XGBoost model. The orbital elements are denoted as follows: The semimajor axis (a), the eccentricity (e), the inclination (i), the argument of pericenter (ω), the longitude of the node (Ω), and the true anomaly (f). Planet-specific parameters are indicated with a subscript p. The mass ratio of the system is represented by μ.

XGBoost, the highest-accuracy algorithm, misclassified only 303 instances, with a nearly balanced distribution of errors between stable and unstable classes (150 and 153 instances, respectively). It also correctly identified the highest number of stable instances (2211). Random Forest and LightGBM achieved the highest precision for the stable class (0.95). They slightly outperformed XGBoost in this metric.

Random Forest showed the best performance in correctly classifying unstable instances (17 536). However, it also had the highest number of false positives, misclassifying 285 stable instances as unstable. This is nearly twice the error rate of XGBoost for this class.

5.2. Feature importance

An interesting outcome of the XGBoost algorithm is the feature importance, which estimates how effectively each feature contributes to reducing the impurity throughout all the decision tree splits. The feature importance is determined by averaging the importance of each feature across all decision trees, and Figure 7 illustrates the relative importance of features in our classification task.

Particle orbital elements dominate the top three positions. The semimajor axis (a), eccentricity (e), and inclination (i) score 0.3567, 0.2783, and 0.2351, respectively. These findings agree with the distribution patterns observed in Section 3, where we noted significant reductions in stable systems beyond $0.6r_{\mathrm{Hill}}$ and the absence of stable particles with inclinations near 90° or eccentricities ≥0.8. Planet eccentricity ranks fourth in importance (0.0564), followed by the particle argument of pericenter (ω, 0.0274). The remaining features (particle true anomaly (f), mass ratio (μ), planet true anomaly (f_{p}), and particle longitude of node (Ω)) contribute less significantly, with scores ranging from 0.0119 to 0.0110.

These results quantify the relative importance of the orbital parameters in determining the system stability that most significantly influence the dynamical outcomes.

6. Comparative analysis

We applied our best ML model, XGBoost, to predict four distinct diagrams with initial conditions ($a \times e$) with the aim of reproducing the results of Domingos et al. (2006), Pinheiro & Sfair (2021), and those for a particular Saturn moon group known as Inuit. In each diagram, the test particles were uniformly distributed within a range of initial semimajor axes from 1.1 r_{p} to

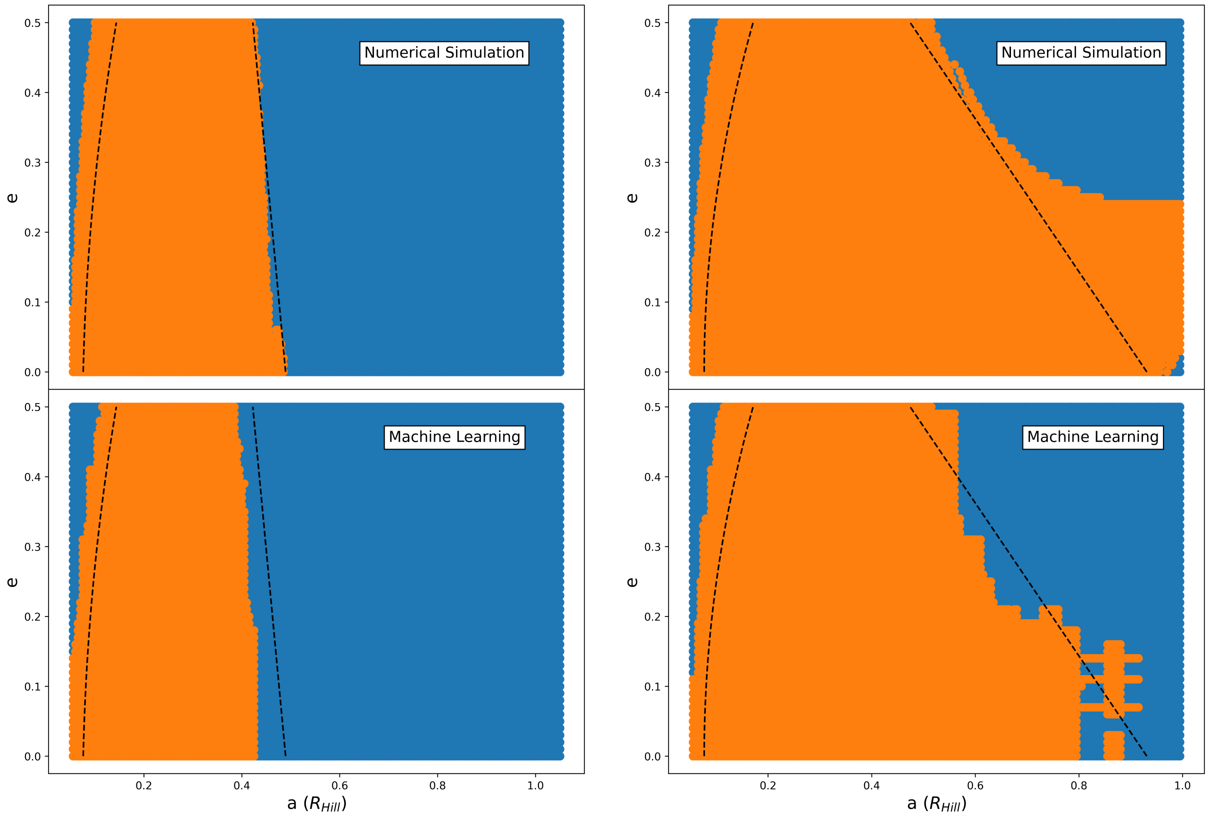


Fig. 8. Comparison of numerical simulations (upper panels) and ML predicted outcomes (lower panels) for stable (orange) and unstable (blue) particles. The initial conditions match those from Domingos et al. (2006) for $\mu = 10^{-3}$, with i = 0° (left) and i = 180° (right). The dashed lines represent the analytical expression from Domingos et al. (2006).

1 $r_{\rm Hill}$ with a step size of $\Delta a = 0.1 r_{\rm p}$, and the initial eccentricities ranged from 0 to 0.5, with $\Delta e = 0.01$.

The XGBoost model, which demonstrated superior performance in our earlier analysis with an accuracy of 98.48%, was tasked with predicting the stability in these specific planetary systems. This approach allowed us to evaluate the generalization capabilities of the model for diverse orbital configurations and to compare its predictions directly with established numerical results.

An entire stability map with ~10 000 different initial conditions with numerical simulations requires between two and four days on a single core of an Intel i7-1165G7 processor. The ML predictions, with the saved training, generated a map in 0.5 seconds.

6.1. Comparison with Domingos et al. (2006)

Domingos et al. (2006) numerically simulated multiple stable maps to delineate the boundaries of stable regions surrounding a close exoplanet orbiting its star at a semimajor axis of 0.1 au. They derived an analytical expression for the critical exosatellite semimajor axis as a function of the eccentricity of the satellite and the planet for two different orbital inclinations 0° and 180°.

We reproduced $(a \times e)$ diagrams of two different systems explored by Domingos et al. (2006), one on a prograde and the other on a retrograde orbit. The mass ratio of the planet and star was defined as $\mu = 10^{-3}$, with a planet radius $r_{\rm p} = 0.05 r_{\rm Hill}$ and orbiting a Sun-like star on a circular orbit.

Figure 8 compares the numerical simulation results with machine-learning predictions. The left panels represent the results for a prograde system, and the right panels show retrograde systems. The performance of our model predictions are summarized in Figure 9, where the dashed lines refers to a prograde system and the solid lines to the retrograde case. The dashed lines represent the analytical expression derived by Domingos et al. (2006).

In the two scenarios, the model accurately classified the main structure of the stable region (orange points), but it showed some discrepancies at the boundaries. For the prograde case, the model closely replicated the stable region, with only minor differences at the edges. The stable region extends to about 0.49 $r_{\rm Hill}$ in the numerical and ML results, which agrees well with Domingos et al. (2006). The model achieved an overall accuracy of 95.86%, a precision of 94% for unstable particles and 100% for stable particles, a recall of 100% for unstable particles and 89% for stable particles, and an F1-score of 97% for unstable particles and 94% for stable particles.

The performance comparison of the analytical expression for the critical semimajor axis proposed by Domingos et al. (2006) and our ML model revealed similar results overall. While the equations of Domingos et al. (2006) exhibit a marginally higher accuracy (96.05%) and recall for the stable and unstable classes (91.80 and 98.70%, respectively), our ML model outperforms their results in precision for both classes. Specifically, Domingos et al. (2006) achieved a precision of 97.78% for stable particles and 95.08% for unstable particles, whereas our model achieved

Pinheiro, T. F. L. L., et al.: A&A, 693, A246 (2025)

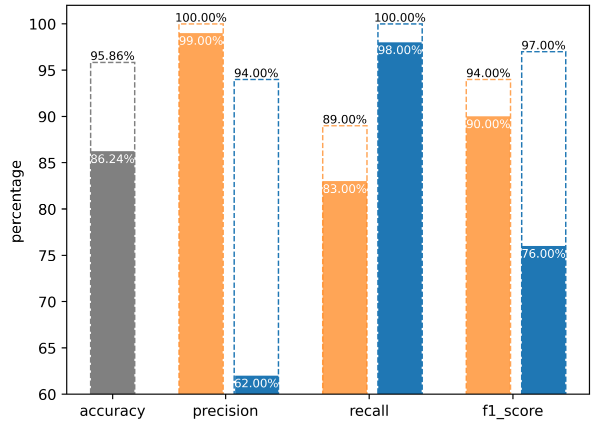


Fig. 9. Performance comparison of numerical simulations and machine-learning predictions for prograde (dashed line) and retrograde (solid line) orbits from Domingos et al. (2006). The blue and orange regions represent unstable and stable particles, respectively.

a precision of 100% in both classes. The F1-score for the two methods has the same percentage.

For the retrograde case, the model captured the overall larger stable region, extending to about 0.93 r_{Hill}. However, it slightly underestimated the stable region extent for eccentricities above 0.3. The ML model also missed some stable particles with semimajor axes beyond 0.8 r_{Hill}, in particular, at lower eccentricities.

Despite these limitations, the ML model achieved an accuracy of 86.24%. It outperformed the analytical expression by Domingos et al. (2006), which had an accuracy of 85.04%. Furthermore, the ML model attained a precision of 99% for stable particles and 62% for unstable particles, a recall of 83% for stable particles and 98% for unstable particles. These results yielded F1-scores of 90% for stable particles and 76% for unstable particles. In comparison, the equations of Domingos et al. (2006) achieved slightly lower F1-scores, with 89.26% for stable particles and 74.85% for unstable particles, along with a recall of 80.81% for the stable class and a precision of 59.94% for the unstable class. However, their precision and recall for stable and unstable particles were marginally better, with values of 99.83 and 99.54%, respectively.

These results demonstrate that our model model can accurately predict the general structure of stable regions for prograde and retrograde orbits, but areas for potential improvement are also highlighted, in particular, in capturing fine details at stability boundaries and for highly eccentric orbits in retrograde systems. Even the analytical expression by Domingos et al. (2006), which only applies to $\mu = 10^{-3}$ and inclinations of 0° or 180° failed to classify many stable particles beyond the black line. A further refinement of the model, possibly through an expanded dataset or enhanced feature engineering, could address these minor discrepancies.

6.2. Comparison with Pinheiro & Sfair (2021)

Pinheiro & Sfair (2021) studied the stability of a possible ring system around the candidate planet PDS110b. They ran 1.3×10^6 numerical simulations of the three-body problem, each representing a different initial condition. We evaluated the performance of our model for the PDS110b system, specifically

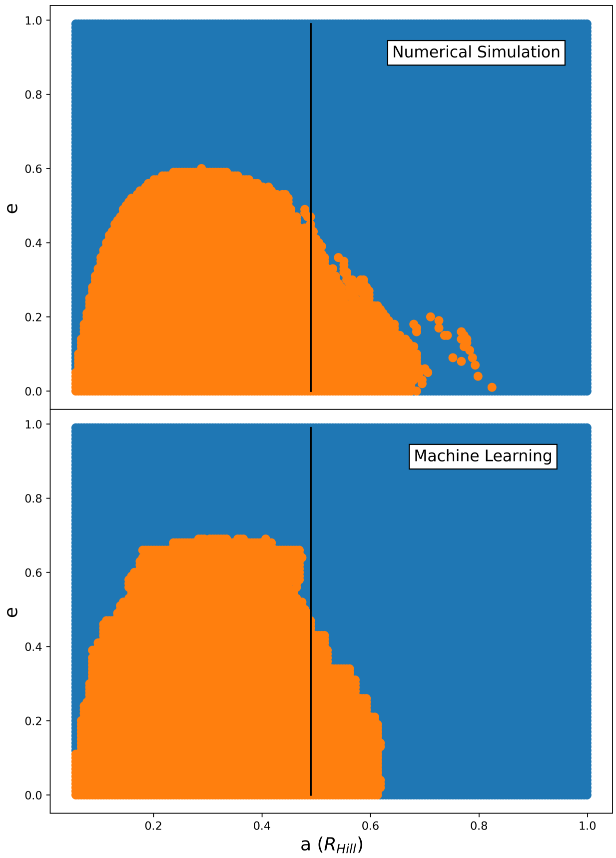


Fig. 10. Stability maps for PDS110b. Top panel: numerical simulation results. Bottom panel: ML prediction. The black line represents the minimum radial size for the observed ring as reported by Pinheiro & Sfair (2021).

considering a scenario in which the planet mass was equivalent to 6.25 Jupiter masses, the eccentricity was 0.2, and the ring inclination was 150°. This scenario corresponds to one of the possible parameters for a retrograde ring system around PDS110b.

Figure 10 shows the numerical and predicted stability maps around PDS110b. Although this system is retrograde, the model performed much better than in the previous example in this case. It achieved an accuracy of 93.68% over ~20 000 predicted initial conditions. The number of stable and unstable instances was almost equivalent, with 5722 actual stable and 14 373 actual unstable instances, respectively.

Figure 11 shows precision, recall, and F1 scores for both classes. The percentages range between 85 and 98%. The slight difference is attributed to the model generalization at the boundary of the stable region, where it misclassified 286 instances as unstable and 984 instances as stable. Some of these misclassifications belong to outlier particles beyond the stable region. Our numerical simulations covered a time span of 10^4 orbital periods; some of these stable particles might become unstable with longer integration times, however.

6.3. Stability map of the Saturn Inuit satellites

Some irregular Saturn moons are classified into three different groups that are distinguished by their inclination range. We

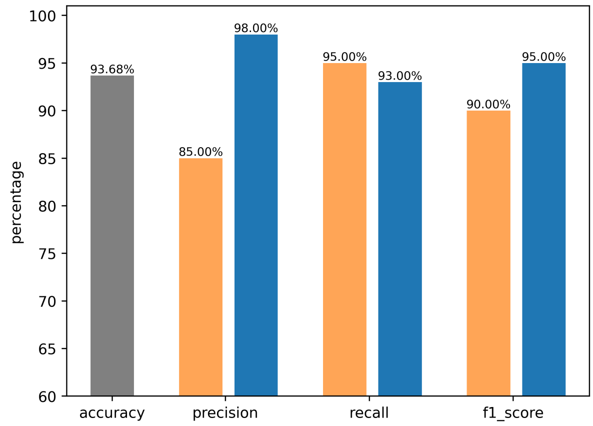


Fig. 11. Performance metrics for the PDS110b system prediction.

Table 2. Orbital elements of Saturn satellites belonging to the Inuit group.

Moon	a (r_{Hill})	e	i (°)
Kiviuq	0.1803	0.334	45.71
Ijiraq	0.1805	0.316	46.44
S/2019 S1	0.1825	0.623	44.38
Paaliaq	0.2466	0.364	45.13
Siarnaq	0.2844	0.295	45.56
Tarqeq	0.2922	0.160	46.09
S2428b	0.2835	0.472	44.43
T522499	0.2824	0.242	48.11

Notes. Some recently discovered Inuit satellites were not included in Table 2 as their orbital eccentricities are not well determined. References: https://carnegiescience.edu/

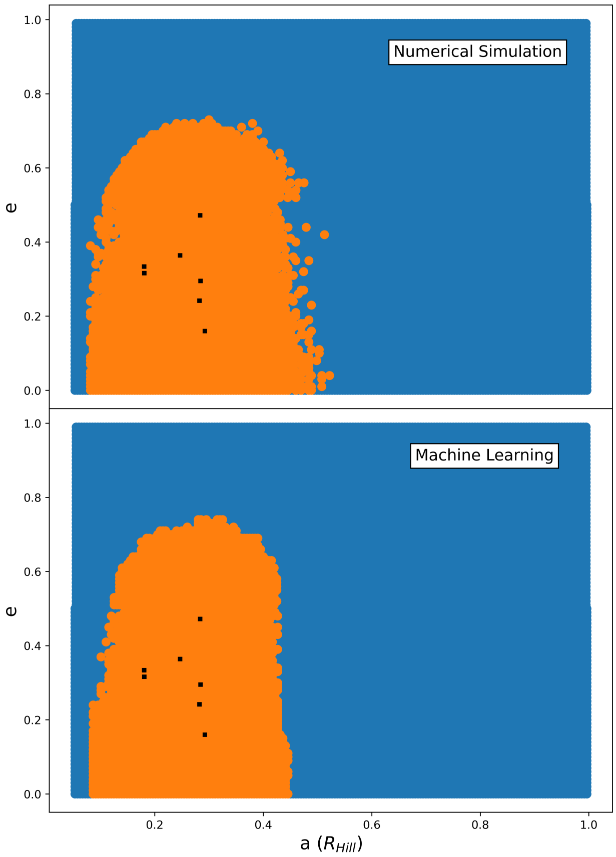


Fig. 12. Stability maps for the Saturn Inuit satellites. Top panel: Numerical simulation results. Bottom panel: ML prediction. The black squares represent the actual satellites belonging to the Inuit group.

generated a stability map for the Inuit group, where the range of inclination was between 40° and 50°. Table 2 summarizes the orbital elements of the moons belonging to this group.

As another test, we ran ~20 000 numerical simulations of a three-body problem representing the Sun, Saturn (corresponding to $\mu = 2.857 \times 10^{-4}$), and a test particle. The initial orbital inclination of the test particle was set to 45°, and the argument of pericenter, true anomaly, and longitude of node were randomly selected from 0° to 360° using a uniform distribution.

Figure 12 shows the numerical and machine-learning predictions for this example. The black squares represent the satellites belonging to the Inuit group.

This example produced our best performance with an accuracy of 97.57%, along with a precision and recall for the stable class of 97 and 93%, respectively, and with a precision and recall for the unstable class of 98 and 99% (Fig. 13). In total, the ML model incorrectly classified only 470 instances out of ~20 000, and most of them were spread particles at the boundary of the stable region. This good result is reflected in the F1-score, which for both classes reached high values, 95% for the stable class and 98% for the unstable one.

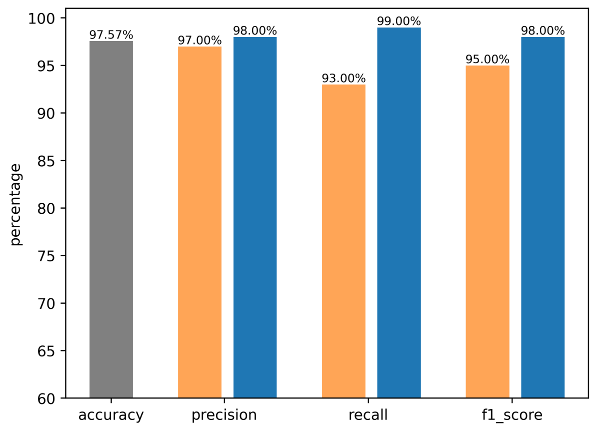


Fig. 13. Performance metrics for the Saturn's Inuit satellites prediction.

7. Final remarks

This study demonstrates the efficacy of ML techniques in predicting orbital stability for a hypothetical planet. We used a comprehensive dataset derived from 10^5 numerical simulations of three-body systems that encompassed a wide range of orbital and physical parameters. These parameters included the mass ratio of the system, the semimajor axis, the inclination, the argument of pericenter, the longitude of the node, the eccentricity, and the true anomaly for the planet and test particles.

Our numerical simulations revealed that 11.83% of the particles remained stable throughout the integration period, while

47.22% collided with the planet and 40.95% were ejected from the system. This imbalanced class distribution necessitated the application of resampling methods to enhance the model performance. We evaluated five ML algorithms: Random Forest, Decision Tree, XGBoost, LightGBM, and Histogram Gradient Boosting, and all algorithms demonstrated a comparable accuracy performance. Through rigorous hyperparameter tuning and threshold optimization, we identified XGBoost as the best-performing model. It achieved an accuracy of 98.48%. To validate the generalization capabilities of our model, we applied it to reproduce the numerical results of Domingos et al. (2006), Pinheiro & Sfair (2021), and a stability map for the Saturn Inuit group of satellites. The model demonstrated a robust performance and achieved an accuracy of 97.57% for these diverse scenarios. Even though our dataset covered inclinations from 0° to 180°, near i = 180° the stable region exhibited significant sensitivity to minor inclination variations, which resulted in a decreased performance. Additionally, we also tested the performance of our model in the case of Earth ($\mu = 3{\times}10^{-6}$) and obtained an accuracy of 95.17%.

Our results show the potential of ML algorithms as powerful tools for reducing computational time in three-body simulations in order 10^5 times faster than traditional numerical simulations. The ability to generate stability maps that cover a wide range of orbital and physical parameters within seconds represents a significant advancement in efficiency compared to traditional numerical methods. The implementation of our model will be made accessible through a public web interface, facilitating its use by the broader scientific community.

As future work, we are refining the model performance, in particular in capturing fine details at the stability boundaries and for highly eccentric orbits and highly retrograde system.

Additionally, we plan to expand the dataset to include more diverse planetary system configurations that might enhance the generalization capabilities of the model and broaden its applicability to a wider range of astrophysical scenarios.

Acknowledgements. We would like to thank the anonymous referee for the constructive comments that greatly improved the manuscript. This research was financed in part by: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001; Fundação de Amparo a Pesquisa no Estado de São Paulo (FAPESP) - Proc. 2016/24561-0; German Research Foundation (DFG) - Project 446102036.

References

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. 2021, Artif. Intell. Rev., 54, 1937
Breiman, L. 1996, Mach. Learn., 24, 123
Breiman, L. 2001, Mach. Learn., 45, 5
Carruba, V., Aljbaae, S., Caritá, G., et al. 2023, Front. Astron. Space Sci., 10, 1196223
Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, J. Artif. Intell. Res., 16, 321
Chen, T., & Guestrin, C. 2016, in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785
Chen, P.-W., Wang, J.-Y., & Lee, H.-M. 2004, IEEE, 13 2035
Cranmer, M., Tamayo, D., Rein, H., et al. 2021, PNAS, 118, e2026053118
Cutler, A., Cutler, D., & Stevens, J. 2011, Random Forests, 45, 157
Domingos, R. C., Winter, O. C., & Yokoyama, T. 2006, MNRAS, 373, 1227
Friedman, J. H. 2001, Annal. Stat., 29, 1189
Fushiki, T. 2011, Stat. Comput., 21, 137
Géron, A. 2022, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow (USA: "O'Reilly Media, Inc.")
Han, H., Wang, W.-Y., & Mao, B.-H. 2005, in International Conference on Intelligent Computing (Berlin: Springer), 878
He, H., Bai, Y., Garcia, E. A., & Li, S. 2008, in International Joint Conference on Neural Networks, IEEE, 1322
Holman, M. J., & Wiegert, P. A. 1999, AJ, 117, 621
Hunter, R. 1967, MNRAS, 136, 245
Ibrahim, M., AbdelRaouf, H., Amin, K. M., & Semary, N. 2023, Int. J. Comp. Inform., 10, 36
Ke, G., Meng, Q., Finley, T., et al. 2017, Adv. Neural Inf. Process. Syst., 30, 3146
Kumar, M., & Sheshadri, H. 2012, Int. J. Comp. Appl., 44, 1
Lam, C., & Kipping, D. 2018, MNRAS, 476, 5692
Mitchell, T. 1997, Machine Learning, (McGraw Hill)
Mohammed, R., Rawashdeh, J., & Abdullah, M. 2020, in International Conference on Information and Communication Systems, IEEE, 243
Neto, E. V., & Winter, O. 2001, AJ, 122, 440
Pinheiro, T. F., & Sfair, R. 2021, A&A, 652, A149
Quinlan, J. R. 1986, Mach. Learn., 1, 81
Rein, H., & Spiegel, D. S. 2015, MNRAS, 446, 1424
Rieder, S., & Kenworthy, M. A. 2016, A&A, 596, A9
Schneider, J., Dedieu, C., Le Sidaner, P., Savalle, R., & Zolotukhin, I. 2011, A&A, 532, A79
Shalev-Shwartz, S., & Ben-David, S. 2014, Understanding Machine Learning: From Theory to Algorithms (Cambridge: Cambridge University Press)
Stehman, S. V. 1997, Remote Sens. of Environ., 62, 77
Tamayo, D., Silburt, A., Valencia, D., et al. 2016, ApJ, 832, L22
Tamayo, D., Cranmer, M., Hadden, S., et al. 2020, PNAS, 117, 18194
Wade, C., & Glynn, K. 2020, Hands-On Gradient Boosting with XGBoost and Scikit-learn: Perform Accessible Machine Learning and Extreme Gradient Boosting with Python (India: Packt Publishing Ltd)
Weerts, H. J., Mueller, A. C., & Vanschoren, J. 2020, arXiv e-prints [arXiv:2007.07588]
Zheng, Z., Cai, Y., & Li, Y. 2015, Comp. Inform., 34, 1017
Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. 2016, Big Data Res., 5, 2

Appendix A: Machine learning algorithms

Here we present the main concepts and characteristics of the tree-based ML algorithms used in this paper.

Decision Tree

Decision Tree, first introduced by Quinlan (1986), classifies instances by splitting them into leaf nodes (Mitchell 1997). This method builds a tree formed by a sequence of several nodes, each with a specific rule ($x_i < t_i$), where t_i is a threshold value of a single feature (x_i) in the i^{th} node (Shalev-Shwartz & Ben-David 2014). The tree starts with the root node, where the data are split into two new subsets and nodes, which can be either a leaf node or a decision node. The leaf or terminal node is where the instance is classified, while the decision node splits the sample again into two new subsets and nodes. This process repeats until the end of the branch.

The choice of feature and best threshold value for each node depends on the impurity of the subsequent two subsets, which can be calculated by estimating the class distribution before and after the split (Géron 2022).

Bagging classifier: Random Forest

Generating a set of classifiers can improve accuracy and robustness. When this approach uses the same training algorithm for every predictor to train different random subsets, the method is called bagging (Géron 2022). The Bagging classifier creates random subsets of the dataset and trains a model on each subset (Breiman 1996). This method returns a combination using a voting scheme of predictions from all predictors.

The bagging algorithm for trees works by iteratively selecting a subset of instances from a training set through bootstrapping, then fitting a tree to these selected instances. Predictions for unseen instances are made by averaging the predictions from all individual trees (Cutler et al. 2011).

Random Forest, introduced by Breiman (2001), is an example of a bagging classifier and an ensemble of Decision Trees. Random Forest reduces the risk of overfitting by decreasing the correlation between trees (Shalev-Shwartz & Ben-David 2014). It trains numerous decision trees using different subsets of the training data. The final prediction is determined by the most voted outputs among these individual decision trees.

Boosting classifiers

Boosting aims to improve the accuracy of any given learning algorithm. This method combines several weak learners into a strong learner and trains predictors sequentially, with each one attempting to rectify the errors of its predecessor (Géron 2022). It uses the residuals of the previous model to fit the next model. The following subsections present an overview of XGBoost, LightGBM, and Histogram Gradient Boosting.

XGBoost

XGBoost (Extreme Gradient Boosting) is a decision tree ensemble developed by Chen & Guestrin (2016) based on the idea of Gradient-Boosted Tree (GBT) by Friedman (2001). It computes the residual of each tree prediction, which are the differences between the actual values and the model predictions. These residuals update the model in subsequent iterations, reducing the overall error and improving predictions (Wade & Glynn 2020). Unlike sequentially built Gradient-Boosted Trees, XGBoost splits the data into subsets to build parallel trees during each iteration.

XGBoost differs from Random Forest in its training process. It incorporates new trees predicting the residuals of previous trees, and combines these predictions, assigning varying weights to each tree, for the final prediction (Chen & Guestrin 2016).

LightGBM

Light Gradient Boosting Machine (or LightGBM) is a variant of the Gradient-Boosted Tree released in 2016 as part of Microsoft's Distributed Machine Learning Toolkit (DMTK) project (Ke et al. 2017). The method employs histograms to discretize continuous features by grouping them into distinct bins. LightGBM introduces the leaf-wise growth strategy, which converges quicker and achieves lower residual.

LightGBM provides different characteristics compared to XGBoost, offering alternative benefits that may be more suitable depending on the specific use case. These include improved memory usage, reduced cost of calculating the gain for each split, and reduced communication cost for parallel learning, resulting in lower training time compared to XGBoost. The algorithm also uses the new Gradient-Based One-Sided Sampling (GOSS) and Unique Feature Bundle (EFB) techniques (Ke et al. 2017). GOSS creates the training sets for building the base trees, while EFB groups sparse features into a single feature (Bentéjac et al. 2021).

Histogram Gradient Boosting

Histogram Gradient Boosting is a boosting model inspired by LightGBM. For training sets larger than tens of thousands of instances, this technique, with its histogram-based estimators, proves very efficient and faster than previous boosting methods (Wade & Glynn 2020). Traditional decision trees require long processing times and heavy computation for huge data samples, as the algorithm depends on splitting all continuous values and characteristics (Ibrahim et al. 2023).

Histogram Gradient Boosting streamlines this process by using histograms to bin the continuous instances into a constant number of bins. This approach reduces the number of splitting points to consider in the tree, allowing the algorithm to leverage integer-based data structures (Wade & Glynn 2020). Consequently, this improves and speeds up the tree implementation.

Appendix B: Imbalanced class

Random oversampling involves replicating instances randomly within the dataset to achieve a balance in class distribution. This approach may lead to overfitting, and an alternative solution to avoid this is to use the SMOTE.

Chawla et al. (2002) proposed oversampling the minority class by generating "synthetic" instances. Firstly, this method calculates the difference in feature vectors between an instance and its nearest neighbors. In the second step, it multiplies this difference by a random number ranging from 0 to 1, and a new instance is subsequently created by adding this result to the features of the original instance.

Another oversampling technique based on SMOTE is Borderline-SMOTE (Han et al. 2005). This method attempts to identify the borderline of each class and avoids using the outliers (noise points) of the minority class to resample them. The algorithm works with the following steps:

1. For each instance of the minority class, check which class its nearest neighbors belong to.
2. Count the number of nearest instances that belong to the majority class.
3. If all neighbors are from other classes, this point is classified as a noise point, and it is ignored when resampling the data.
4. If more than half of the neighbors are from other classes, this is a border point. In this case, the algorithm identifies the K nearest neighbors that are of the same class and generates a synthetic instance between them.
5. If more than half of the neighbors are from the minority class, it is a safe point and the SMOTE technique is applied normally.

The last oversampling approach implemented in this work is ADASYN, which generates synthetic instances for the minority class by considering the weighted distribution of this particular class.

The methodology employed by He et al. (2008) involves an estimation of the impurity of the nearest neighbors r_i for each instance in the minority class, given by

$$r_i = \frac{\Delta_i}{K} \tag{B.1}$$

where Δ_i is the number of non-minority and K the number of neighbors.

The subsequent step normalizes the r_i value as

$$\hat{r}_\mathrm{i} = \frac{r_\mathrm{i}}{\sum\limits_{i=1}^{m} r_\mathrm{i}} \tag{B.2}$$

where m is the size of the minority class data. This normalized value is then multiplied by the total number of synthetic instances to be generated. This procedure proportions the number of synthetic instances to be created for each minority instance.