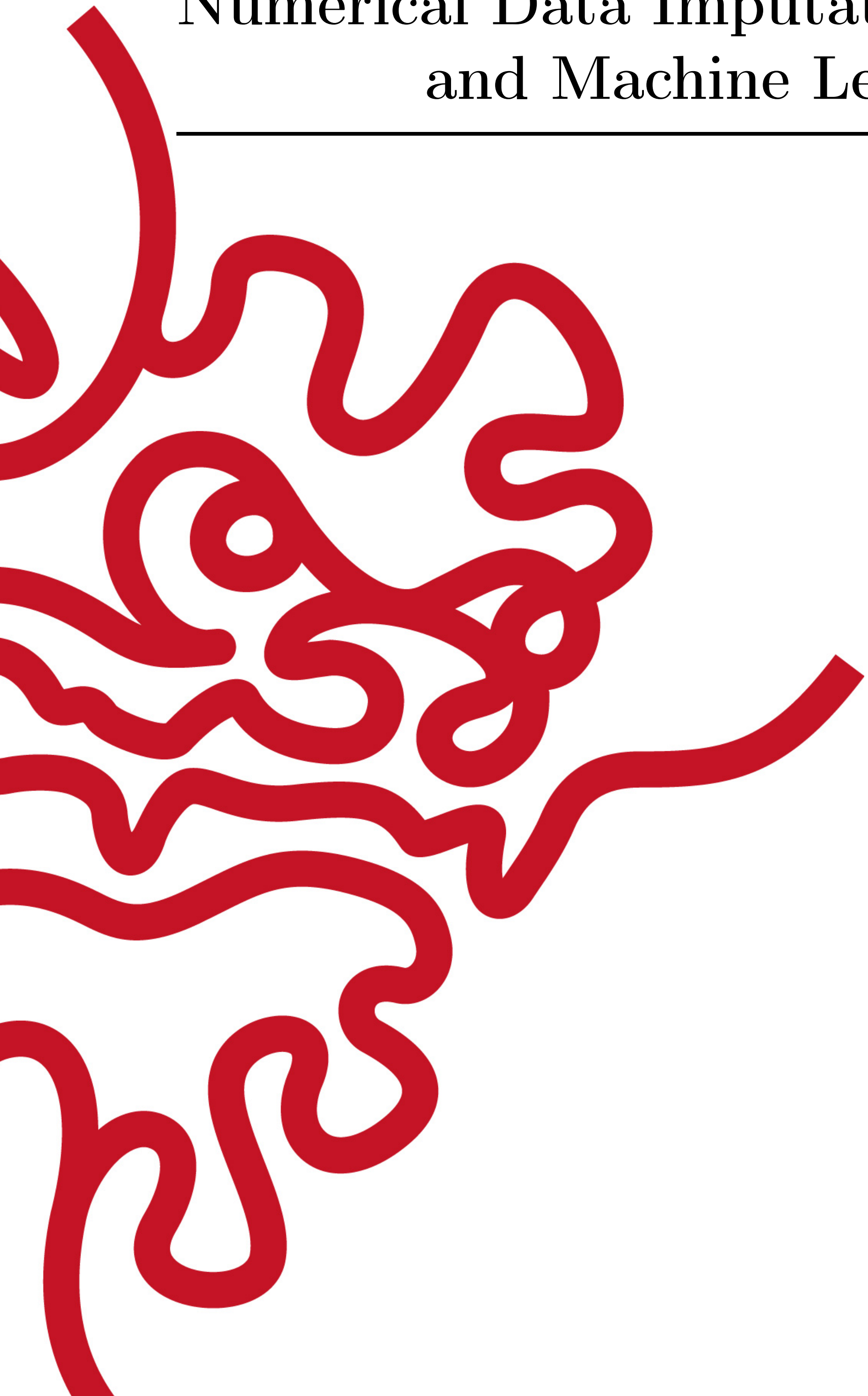
Okinawa Institute of Science and Technology
Graduate University

Thesis submitted for the degree

Doctor of Philosophy

Planetary Systems Insights through Numerical Data Imputation Algorithms and Machine Learning



by

Florian Lalande

Supervisor: **Kenji Doya**

March 2024

Declaration of Original and Sole Authorship

I, Florian Lalande, declare that this thesis entitled *Planetary Systems Insights through Numerical Data Imputation Algorithms and Machine Learning* and the data presented in it are original and my own work.

I confirm that:

- No part of this work has previously been submitted for a degree at this or any other university.
- References to the work of others have been clearly acknowledged. Quotations from the work of others have been clearly indicated, and attributed to them.
- In cases where others have contributed to part of this work, such contribution has been clearly acknowledged and distinguished from my own work.
- None of this work has been previously published elsewhere, with the exception of the following:

1. **Lalande Florian** and Doya Kenji. *Numerical Data Imputation: Choose kNN Over Deep Learning*, Similarity Search and Applications, Lecture Notes in Computer Science, Volume 13590, Springer, 2022.

   Contributions – FL gathered data, developed the code, performed the analysis, presented the results, and wrote the manuscript. KD was in charge of the project administration, supervision, and funding. Both authors conceptualized the work.

2. **Lalande Florian** and Doya Kenji. *Numerical Data Imputation for Multimodal Data Sets: A Probabilistic Nearest-Neighbor Kernel Density Approach*, Transactions on Machine Learning Research (Reproducibility Certification), 2023. ISSN 2835-8856.

   Contributions – FL gathered data, developed the new imputation method, conducted benchmark, presented the results, and wrote the manuscript. KD was in charge of the project administration, supervision, and funding. Both authors conceptualized the work.

3. **Lalande Florian** and Trani Alessandro Alberto. *Predicting the Stability of Hierarchical Triple Systems with Convolutional Neural Networks*, The Astrophysical Journal (ApJ) 938, 2022. ISSN 0004-637X.

Contributions – FL generated data using a package previously developed by AAT. FL developed the software, performed experiments, and collected results. AAT supervised the project and was in charge of the funding. Both authors discussed results, and wrote the manuscript. Both authors conceptualized the work.

4. **Lalande Florian**, Yoshitomo Matsubara, Naoya Chiba, Tatsunori Taniai, Ryo Igarashi, and Yoshitaka Ushiku. *A Transformer Model for Symbolic Regression towards Scientific Discovery*, accepted as Oral and Poster at NeurIPS 2023 AI for Science Workshop.

   Contributions – FL wrote the code, generated data, developed the software, ran experiments, collected results, and analyzed them. YU was in charge of the funding. YU , RI, YM, and NC were in charge of the supervision of the project. All authors conceptualized the work.

Date: March 2024
Signature:

Lalande

Abstract

Since the first discoveries in the early 1990's, the number of known exoplanets has exploded to reach over 5,500 as of December 2023. But the recorded information for each planets is sparse, with a lot of missing values, preventing from confidently drawing overarching conclusions. As most traditional data imputation methods provide a point estimate, they fail at capturing the complexity of multimodal data distributions and provide unreliable estimates in scenarios where data exhibits multiple modes. This calls for a new paradigm to model rich or complex numerical datasets.

This PhD thesis introduces the kNN×KDE, a numerical imputation tool which combines the flexibility of the k-nearest neighbors (kNN) and the simplicity of Kernel Density Estimation (KDE) to model the multi-dimensional distribution of missing data in datasets characterized by multimodality. This new method is tested against traditional and novel data imputation algorithms, and I show that the kNN×KDE not only provides better estimates, but also facilitates their interpretation.

To demonstrate the practical significance of the kNN×KDE, I apply it to the NASA Exoplanet Archive – a dataset riddled with missing values, including both planetary radius and mass, and marked by pronounced multimodality. The analysis of the estimated distributions provided relevant insights into the demographics of the Exoplanet Population, potentially helping future missions to select interesting targets.

In addition, this PhD work includes two artificial neural network applications for planetary system analysis: a Convolutional Neural Network (CNN) to predict planetary system stability and a Graph Neural Network (GNN) to rediscover Newton's Law of Gravitation and attempt to reproduce the scientific discovery of Neptune. Finally, this thesis features a Transformer model for Symbolic Regression applied to 120 real-world physics equations. These additional tools contribute to further characterize planetary systems evolution and understand the limits of Machine Learning for scientific discovery.

Acknowledgment

I would like to begin by thanking my two PhD supervisors, Kenji Dōya and Elizabeth Tasker. You are like super heroes for me, and I admire both of you very much, each in your own way. Kenji, thank you for believing in me, giving me the freedom to pursue astrophysics in your lab, supporting me in my personal development, and always providing me with incredibly sharp and smart ideas at crucial times throughout my PhD journey. Elizabeth, I am very grateful for your patience, your pedagogy, and your communication skills, which helped me learning a lot about exoplanets. I loved very much working closely with you towards the end of my PhD. Thank you!

Next, I want to express my gratitude to my collaborators at the University of Tōkyō: Alessandro Alberto Trani and Yasushi Sutō. I had a lot of fun working with you, and I am happy to count you as friends now. To you Aless, I feel very fortunate to have met you in the beginning of my PhD. I am indebted to you for taking care of me when my proposal was still undecided. Working with you was very salvaging! Thank you very much.

To my friends in Okinawa, in France, and all around the world: thank you for your unconditional support, even when you guys don't have a single clue of what I'm doing. To my OIST fellows, thank you for brightening my time in Okinawa, for the trips and the good times, and for heating up my lunches with political or scientific debates. I loved it all! To Hend, my partner in crime, I am immensely proud and thankful to have met you. I don't think I'd have survived at OIST without you. Thank you for being who you are. I will miss you.

À ma famille : merci d'être toujours aussi présents malgré la distance. Pour croire en moi quand je doute. Pour me donner le courage de persévérer dans mes aventures. Papy et Mamie, merci infiniment pour écouter mes histoires pendant des heures au téléphone et pour vos précieux encouragements. Papa, je sais combien tu es fier de moi, et ça me rend très fier à mon tour. Merci d'être toujours là pour moi. La Foufouna, merci d'être toujours aussi disponible, à l'autre bout de mon téléphone, et de me tenir au courant des événements dans la famille. Je suis tellement content de t'avoir ! Aurélien, mon brother, merci d'être qui tu es, et de bientôt faire de moi le plus heureux des tontons. Et enfin, à la meilleure Maman du monde. Merci pour tout l'amour que tu me portes, et pour toujours croire en moi, quoiqu'il advienne. Je t'aime Maman.

This last paragraph is for the one who suffered the most adversarial effects from this PhD. My Bibi, Kazuki. Thank you so much for always being by my side, for understanding me, for helping me put things into perspective, for treating me like a prince, and for bearing with my stubbornness. I am infinitely grateful for having met you, and very proud to call you my boyfriend. You have made my time as a PhD student much smoother and pleasant than I thought was possible. I am looking forward to our future together. 愛。

List of Publications

Published manuscripts and author contributions are listed in the order of appearance in this thesis.

1. **Lalande Florian** and Doya Kenji. *Numerical Data Imputation: Choose kNN Over Deep Learning*, Similarity Search and Applications, Lecture Notes in Computer Science, Volume 13590, Springer, 2022.

   Contributions – FL gathered data, developed the code, performed the analysis, presented the results, and wrote the manuscript. KD was in charge of the project administration, supervision, and funding. Both authors conceptualized the work.

2. **Lalande Florian** and Doya Kenji. *Numerical Data Imputation for Multimodal Data Sets: A Probabilistic Nearest-Neighbor Kernel Density Approach*, Transactions on Machine Learning Research (Reproducibility Certification), 2023. ISSN 2835-8856.

   Contributions – FL gathered data, developed the new imputation method, conducted benchmark, presented the results, and wrote the manuscript. KD was in charge of the project administration, supervision, and funding. Both authors conceptualized the work.

3. **Lalande Florian** and Trani Alessandro Alberto. *Predicting the Stability of Hierarchical Triple Systems with Convolutional Neural Networks*, The Astrophysical Journal (ApJ) 938, 2022. ISSN 0004-637X.

   Contributions – FL generated data using a package previously developed by AAT. FL developed the software, performed experiments, and collected results. AAT supervised the project and was in charge of the funding. Both authors discussed results, and wrote the manuscript. Both authors conceptualized the work.

4. **Lalande Florian**, Yoshitomo Matsubara, Naoya Chiba, Tatsunori Taniai, Ryo Igarashi, and Yoshitaka Ushiku. *A Transformer Model for Symbolic Regression towards Scientific Discovery*, accepted as Oral and Poster at NeurIPS 2023 AI for Science Workshop.

   Contributions – FL wrote the code, generated data, developed the software, ran experiments, collected results, and analyzed them. YU was in charge of the funding. YU , RI, YM, and NC were in charge of the supervision of the project. All authors conceptualized the work.

Contents

List of Figures

Chapter 1

Introduction – Planetary Systems & Machine Learning

The first Chapter of this thesis provides a general introduction to the fields of interest. While Section 1.1 presents the field of exoplanet research and how exoplanet data is obtained, Section 1.2 gives a brief introduction to Machine Learning (ML) with a specific emphasis on numerical data imputation, a problem intensely studied during this thesis. Next, Section 1.3 introduces the problems to be addressed, lying at the intersection on exoplanetary research and statistics. The outline of this thesis is provided in Section 1.4.

1.1 Planetary Systems and the Hunt for Exoplanets

This first Section presents the current state of exoplanetary science. After providing a quick overview of the field of exoplanet research (Section 1.1.1), this introduction covers the demographics for the confirmed exoplanets population (Section 1.1.2) and presents the advantages of drawbacks of the most prolific detection methods (Section 1.1.3). In the end of this Section, current instrumental limitations, challenges, and potential biases are discussed (Section 1.1.4).

1.1.1 The Exoplanet Revolution

For centuries, our knowledge of planets was limited to the eight within in our Solar System. Despite speculations about the existence of planets around other stars, it remained impossible to affirm or assess how common and how similar to the Earth these planets could be. The breakthrough came in 1995 with the first discovery of an exoplanet - a planet outside the Solar System - orbiting a Sun-like star was officially confirmed [1]. This discovery led to a paradigm shift in astronomy, revealing that distant stars also host their own unique worlds.

Almost 30 years later, we now know of planets orbiting a wide variety of stars, which suggests that planet formation is a frequent byproduct of stellar evolution. Considering the hundreds of billions of galaxies in the observable Universe, each of them with an estimated average of hundred million stars [2], it becomes evident that exoplanets are abundant throughout the cosmos. The number of planets might even surpass the number of stars. For example, our own Sun already counts eight planets alone, while Promixa Century, the nearest star to the Sun at 4.2 light years, has two confirmed planets so far [3, 4] with a third disputed candidate [5].

As of July 2023, we have a count of 5,470 confirmed exoplanets, the majority of which being discovered in the last ten years. But in spite of many planetary systems known, our own Solar System remains unique. A planetary system composed of four inner small rocky planets and four outer gas giant planets has not been probed yet. That said, the star Kepler-90 (located 2,800 light-years away from the Earth) is notable for hosting eight planets in a similar configuration to the Solar System. The innermost six planets range from Super-Earth to Mini-Neptune sizes, while the outermost two planets are gas giants. A conspicuous difference with the Solar System is that all these planets orbit Kepler-90 closer than the Earth orbits the Sun, creating tidally-locked resonant chains [6, 7]. There is no confirmed planetary system with nine or more planets to date.

Like the Earth has its own Moon, planets can also have natural satellites. In particular, gas giants within the Solar System are known for hosting dozens of natural satellites as well as ring systems. However, notwithstanding numerous collaborations and efforts, no exomoon (natural satellite that orbits an exoplanet) has been confirmed so far. There exist few reported candidates [8–10], but further evidence will be needed to assess the discovery of the first exomoon ever detected. Exomoons are expected to be eventually confirmed, perhaps even outnumbering exoplanets, but current instruments are not yet sensitive enough to probe small satellites orbiting planets, which themselves orbits distant glaring stars.

Studying distant exoplanets is an important exercise to better understand the Solar System. Planetary systems, including our own, are thought to be a common byproduct of stellar formation. The widely accepted model of the nebular hypothesis [11] suggests that the gravitational collapse of gas progressively leads to the formation of an accretion disk. Over time, the accretion disk differentiates between a protostar at its core, and a protoplanetary disk at the edge. Gas and dust which did not fall into the protostar will eventually merge as accretion continues due to gravitational forces. Rocks, planetesimals, and protoplanets can form close to the protostar. [12]. Investigating extrasolar planetary systems is an opportunity to rewind the history of the Solar System and better understand planetary formation, and the emergence of natural satellites, rings, tectonic activity or magnetic fields [13].

Besides, the search for planets beyond the Solar System has sparked new interests for the search of extraterrestrial life and bio-signature markers. Analyzing the light of an exoplanet during the transit of its star using spectroscopy can provide information on its atmosphere chemical constituents and thermal structure. The study and modeling of exoplanet atmospheres can also lead to a better understanding of the Earth's atmosphere, particularly precious in a time of climate change. The ARIEL space telescope (Atmospheric Remote-sensing Infrared Exoplanet Large survey) scheduled to be launched in 2029 by the European Space Agency will observed at least a thousand planets that transit their host star, in order to study their chemical composition.

Finally, the famous concept of habitable zone of a star defines a circumstellar range of distances where a planet (assumed to have the same atmospheric composition as the Earth) could support liquid water [14, 15]. As it relies too much on the Earth assumption, this concept is criticized within the astrophysics community, and alternative ways to quantify how similar to the Earth an exoplanet could be are being discussed [16]. That being said, habitable zones remain a useful criterion to search for bio-markers of life, and we have good reasons to believe that planets in their habitable zones exist in large number [17]. However, whether life is abundant in the Universe or not remains a fascinating open question. In spite of the Drake Equation speculating about the odds of finding intelligent life in the Cosmos, the SETI Institute (Search for ExtraTerrestrial Intelligence) remains unsuccessful since 1984. Meanwhile, the Rare Earth Hypothesis speculates that the development of complex life and civilization requires improbable combinations, thus only leaving humanity contemplating its own existence [18]

1.1.2 Host Stars and Exoplanets Demographics

This subsection provides an overview of the parameters of interest when studying exoplanets, as well as their missing rate (proportion of missing values) in the NASA Exoplanet Archive. The Interesting features can be split into three categories: the host stars parameters, the parameters that describe the entire system, and the parameters describing individual planets.

– Host Star Parameters –

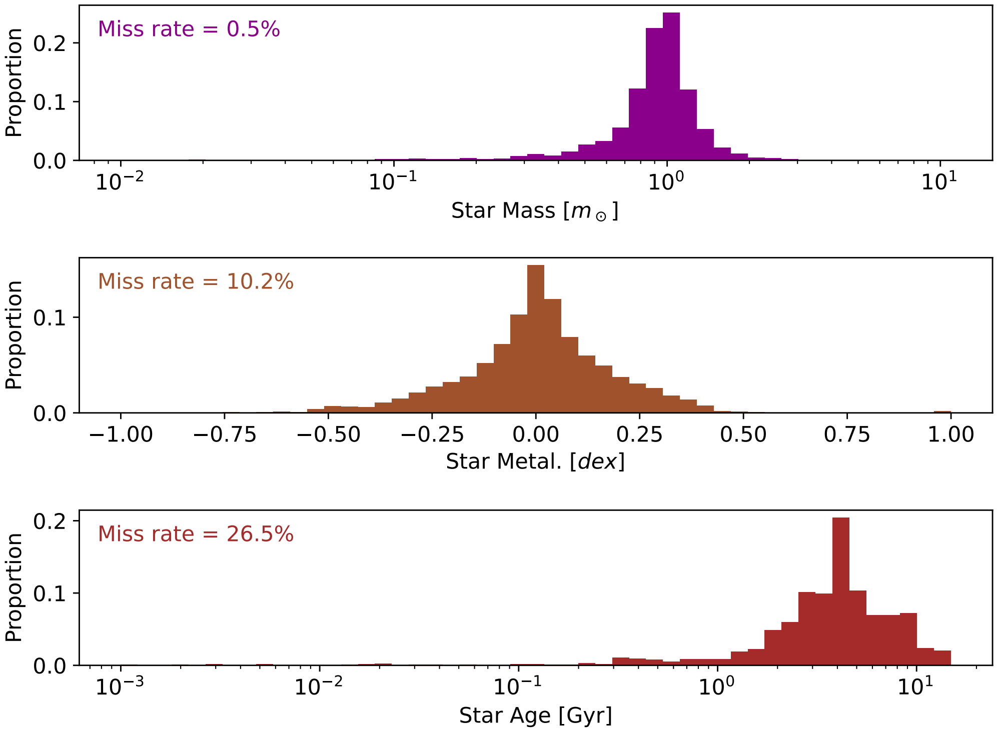


Figure 1.1: Distributions and missing rates for three host star parameters of interest. The missing rates (top right corner) indicate that exoplanets host stars are usually well characterized, with little missing information. Data: NASA Exoplanet Archive (Feb. 2023).

Stars that host confirmed exoplanets can be characterized (like any other star) by their mass, radius, effective temperature, metallicity, or its age. Figure 1.1 shows the distribution for the host stars mass, metallicity, and age of the confirmed exoplanets present in the NASA Exoplanet Archive as of February 2023. Because stars are easier to detect and study than their planets, exoplanets' host star are usually well characterized with low missing rates.

Star mass and radius are expressed in solar units, given by $M_\odot = 1.99 \times 10^{30}\,\mathrm{kg}$ and $R_\odot = 6.96 \times 10^8\,\mathrm{m}$ respectively. The effective temperature of a star informs on its spectral type, i.e. the "colour" of the star, which allows to classify the star in the standard taxonomic stellar classification system [19]. For example, the Sun's effective temperature is $T_{\text{eff.}} = 5,780\,\mathrm{K}$, making it a G-type star with a yellow-white colour.

The star metallicity characterizes the abundance of elements heavier than hydrogen or helium. As most of the baryonic matter in the Universe (i.e. not dark matter) is hydrogen or helium, the word "metal" in astronomy conveniently refers to all elements except hydrogen and helium. To compute the metallicity of a star, one can estimate the fractional mass of hydrogen and helium, respectively denoted as X and Y, and if Z is the fractional mass of all remaining elements, then the metallicity is given by $Z = 1 - X - Y$. Metallicity values are often given in dex, with the metallicity of the Sun $Z_\odot = 0.0122$ as reference [20].

Finally, it is worth mentioning the range of distance where exoplanets have been probed, although this does not intrinsically provide any intrinsic physical information. The distance of a star from us is usually given in parsec or light years (ly), with $1\,\mathrm{pc} = 3.26\,\mathrm{ly} = 3.09 \times 10^6\,\mathrm{m}$. Most of the exoplanets discovered so far are located within 4,000 light years. The closest star from us, Proxima Centauri, is at a distance of 4.2 light years, and the size of the Milky Way is approximately 100,000 light years for comparison [21].

– System Parameters –

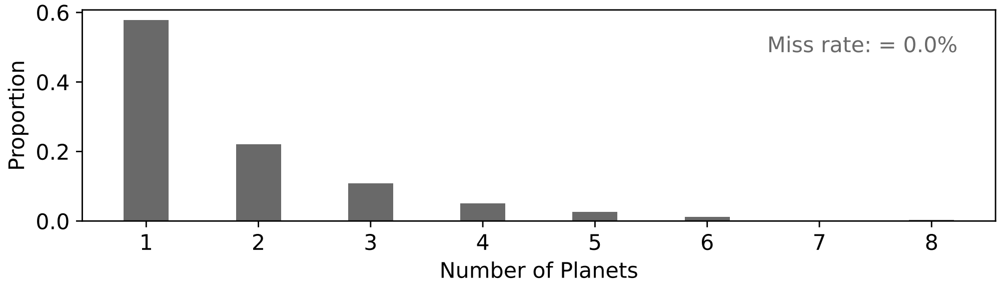


Figure 1.2: Distribution and missing rate for the number of planets per system. Their is obviously no missing data regarding the number of confirmed exoplanet per planetary system. However, this number may be subject to change. Data: NASA Exoplanet Archive (Feb. 2023).

Half-way between host star and planet properties, one might be interested in properties which are inherent to the whole stellar system, like the number of confirmed planets in a system. The number of planets in our Solar System is 8, and the planetary system of

Kepler-90 is the only other one known for also having 8 planets [6]. Figure 1.2 shows the distribution for the number of confirmed exoplanets per system. Unsurprisingly, systems with more planets are harder to probe, and most planetary systems known to date have only a single confirmed exoplanet.

Note that the missing rate of 0 % for the number of known planets in the system can be misleading. Indeed, as soon as an exoplanet is discovered, the number of known planets in that system immediately becomes 1, and therefore cannot be missing. However, this number might be inaccurate and change in the future. This often happens when new planets are discovered in that system, and much more rarely can also happen when previously confirmed planets are reexamined and classified as false positive (RIP, Pluto). Eclipsing binary stars can typically mimic transiting planet signals [22].

It is also worth noting that some planets can also orbit binary star systems. This is the case for the system of Kepler-47 (about 3,400 light years away), composed of two stars – Kepler-47A and Kepler-47B – with a planetary system of 3 planets – Kepler-47b, Kepler-47c, and Kepler-47d) [23]. Such cases, referred to as circumbinary planets, have greatly challenged planet formation models [24].

– Individual Planet Parameters –

This paragraph finally presents the characteristics of individual planets. Figure 1.3 presents the distributions and the missing rates for five essential parameters used to describe exoplanets. Because of possible exoplanet detection methods, the mass and/or the radius of an exoplanet cannot always be measured. This leads to a high missing rate for the confirmed exoplanet masses (72.8 %), and a moderately high missing rate for the radius (30.4 %). This problem will be addressed and discussed in Chapter 4.

Of highest relevance, the radius and the mass of exoplanets are important features to be measured. For convenience, they are usually expressed in units of Earth or Jupiter radii and masses, given by $R_\oplus = 6.37 \times 10^3\,\text{km}$ and $M_\oplus = 5.97 \times 10^{24}\,\text{kg}$ for the Earth, and $R_J = 6.99 \times 10^4\,\text{km}$ (approx. $11\,R_\oplus$) and $M_J = 1.90 \times 10^{27}\,\text{kg}$ (approx. $320\,M_\oplus$) for Jupiter. When both the radius and the mass of an exoplanet are known, the bulk density of the planet can be computed, which allows to assess whether the planet is more likely to be gaseous, liquid, icy, or rocky. The bulk density of a planet (or its mean density) only informs on the global composition, but not on the internal structure which may greatly vary – like the interior of Jupiter with a dense core at its center, a surrounding layer of metallic hydrogen, and an outer atmosphere with mostly molecular hydrogen [25].

Parameters that characterize the orbit of an exoplanet are also of particular interest. The orbital period, measured in days or in years, quantifies the time it takes the planet to revolve around its host star. Due to technical reasons (see next Section) and time constraints, most of the planets probed so far have a small orbital period. As can be seen on the third panel of Figure 1.3, the vast majority of the confirmed exoplanets up to date have an orbital period of less than a year, which is only true for two out of the eight planets present in Solar System for comparison.

The orbital inclination is the angle $i \in [0; 180]\,\text{deg}$ between the planet orbital plane and the line of sight for an observer on Earth. As most of the planets have been detected using the Transit Spectroscopy method, their inclination is close to 90 deg. This does not reflect the expected distribution of exoplanet orbital inclination angles, but is merely a

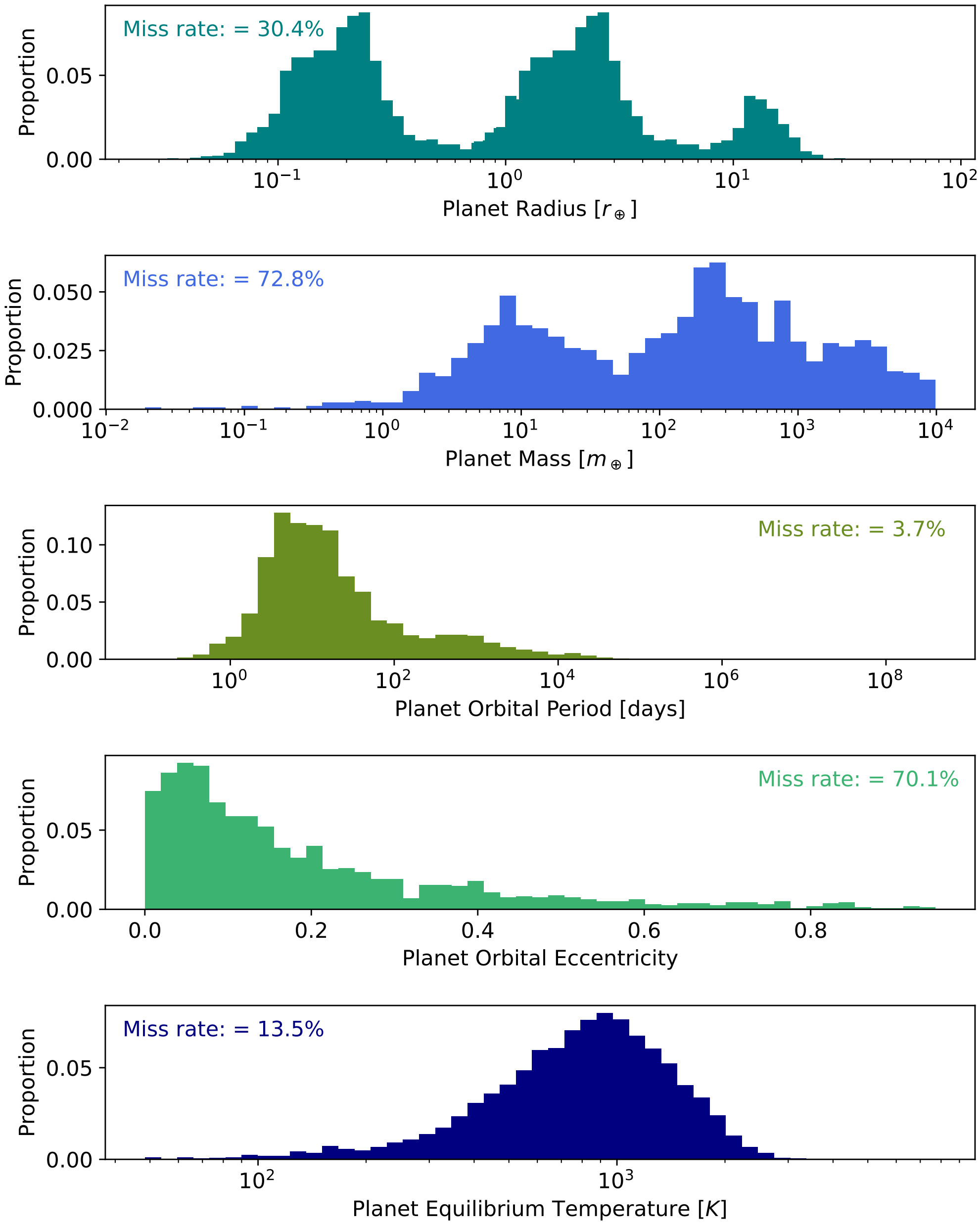


Figure 1.3: Distributions and missing rates for five exoplanet parameters of interest. Although crucial to describe exoplanets, missing rates show that more than two thirds of exoplanet masses remain unknown, and about a third of exoplanet radii are also missing. Data: NASA Exoplanet Archive (Feb. 2023)

bias, result of technical constraints. Finally, the eccentricity $e \in [0; 1]$ characterizes how elliptic the orbit of the planet is. Most exoplanets with low orbital period (less than 20 days) have nearly circular orbits (i.e. $e < 0.1$), thought to be due to tidal circularization, hence not an observational bias this time [26]. But as the high missing rate suggests, it remains technically challenging to measure an exoplanet orbital eccentricity.

1.1.3 Methods of Discovery

Probing planets orbiting distant stars is a technical challenge that requires high resolution instruments. This Section provides an overview of the commonly used exoplanet detection methods and discuss their strengths and weaknesses.

– The Transit Photometry –

By far the most fruitful method, which accounts for approximately 75 % of the total number of discoveries, is the Transit Photometry detection method. If a distant planetary system is nearly perfectly edge-on as seen from the Earth, we are able to observe planets transiting in front of their host star. In the manner of an extremely small eclipse, the transit occludes part of the light we receive from the host star. The period of the shadows allows to infer the orbital period of the planet, and their magnitude allows to estimate the exoplanet radius relatively to its host star. The first exoplanet transit has been observed in 1999, but it was an already known planet at the time: HD-209458 b [27].

The space telescope CoRoT (Convection, Rotation and planetary Transits), operated by the French Space Agency and the European Space Agency from 2006 to 2013, discovered the first transiting exoplanet – CoRoT-7 b – with a density corresponding to a terrestrial planet [28]. Following the discoveries of CoRoT, the NASA Kepler Space telescope has been launched in 2009. It has been continuously monitoring more than 200,000 stars, leading to an explosion in the number of confirmed exoplanets with approximately 2,650 newly added planets [29]. After the Kepler satellite retired in 2018, the new NASA mission TESS (Transiting Exoplanet Survey Satellite) has been taking over its successor, now looking at the entire sky [30]. While Kepler mostly discovered gas giant planets orbiting close to their star, the primary goal for TESS is to detect small rocky planets orbiting around Sun-like stars in our neighborhood.

The Transit Photometry method is easy to scale up and has led to an abundance in exoplanet confirmations. But it has the disadvantage to hope for planet transits in front of their host star, which happens very coincidentally. Supposing that a technologically advanced civilization was looking at our Solar System from a far distance, the probability of a random alignment producing a transit of the Sun by the Earth is only about 0.5 %. Also, this method does not provide measurement for the planet mass, which is a key feature to further characterize the exoplanet. Because of the myriad of planets detected through the Transit method, a lot of confirmed exoplanets present in the NASA Exoplanet Archive have no mass measurement (see Figure 1.3).

– The Radial-Velocity Spectroscopy –

The Radial-Velocity (RV) detection method (or Doppler Spectroscopy method) is historically the first method yielding the discovery of an exoplanet around a Sun-like star in 1995 [1]. Before the inauguration of the Kepler Space Telescope in 2009, the RV method used to be the most productive detection method. Nowadays, the RV method accounts for approximately 20 % of the total number of discoveries.

The idea consists in using powerful spectrographs to indirectly probe the gravitational pulling of a planet on its host star [31]. When a massive enough planet revolves around a host star, the star-planet system is actually in orbital motion around its center of mass.

As a result, it appears to an observer on Earth as if the host star was periodically drawing close and going away from us, which translates into a periodic red-shift / blue-shift in the starlight. The star wobbles can be detected using sensitive enough spectrometers, often at ground-based telescope sites.

A substantial drawback of this method is that it can only measure changes in the star velocity along the line of sight for an observer on Earth. As a consequence, only a measurement of the projected mass along the line of sight is possible, known as the minimum mass and denoted as $M_P \sin(i)$, where M_P is the true (unknown) mass of the planet and i the orbit inclination. In the extreme case where a planet orbits in the plane of the sky (i.e. completely orthogonal to the line of sight), no Doppler effect can be recorded. Also, no radius measurement is provided when using the RV method, leaving the minimum mass the only measured characteristic.

Another limitation of this method lies in the magnitude of the effect it tries to measure. If the target planet is not massive enough comparatively to its host star, no Doppler effect can be observed. That said, current spectrometers can reach high precision, like the HARPS Spectrograph (High Accuracy Radial Velocity Planet Searcher) [32] at La Silla Observatory, or ESPRESSO (Echelle Spectrograph for Rocky Exoplanets and Stable Spectroscopic Observations) [33] at the Very Large Telescope.

– Other Methods –

Few other exoplanet detection methods exist, which altogether account for about 5 % of the total confirmed detections. Among them, the Gravitational Microlensing technique (about 3.6 % of the total) and Direct Imaging (about 1.2 % of the total) are the most significant.

The Gravitational Microlensing technique has allowed for the discovery of about 200 exoplanets. When a star happens to pass in front of another background distant star, the gravitational field of the foreground star produces a lensing effect which magnifies the light of the background star. If the foreground star hosts planets, their own gravitational fields can have a significant contribution to the overall lensing effect. This method is particularly good at detecting dense planets that orbit far from their host star (at least 1 AU away). The Gravitational Microlensing technique is also known for the detection in 2005 of the first exoplanet orbiting a main-sequence star with a mass comparable to the Earth [34].

Finally, and probably the most fascinating detection method, is simply Direct Imaging. However, it is extremely challenging as the reflected light from planets gets easily lost in the glare of their host star. One workaround consists in using coronagraphs to block the light of the host star and conduct infrared observations, where planets emit the most in contrast to their host star. A spectacular example is the planetary system of the star HR 8799, the first to be confirmed via Direct Imaging in 2008 [35]. Figure 1.4 shows the direct imaging of HR 8799 between September 2009 and July 2016, where we can see the displacement of the four exoplanets of this system. Note that unlike with the Transit Spectroscopy method, bets observations for Direct Imaging happen when the planetary system orbital plane aligned with the plane of the sky for an observer an Earth.

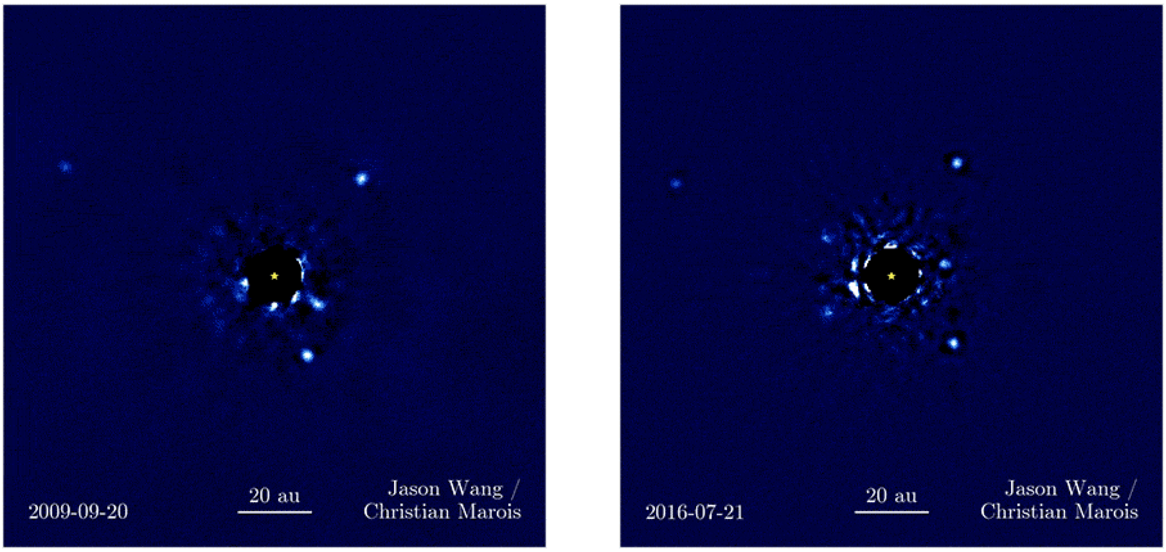


Figure 1.4: Direct Imaging of the planetary system of star HR 8799. The star HR 8799 is located 133 light-years away. Four planets can be identified. They have very long orbital period. Credits: W. M. Keck Observatory and NASA

1.1.4 Current Limits, Challenges, and Biases

Of course, the current state of exoplanetary research is strongly dictated by our technologies and their physical limitations. Observational biases distort our perceptions of the Universe, such that patterns we see within the current population of known exoplanets are most likely not characteristic of the whole (unknown) population of planets. This paragraph discusses the impacts of both technological and physical limitations on the sample of known exoplanets.

– Technical Limits –

Planets detected through the Transit Photometry method account for approximately 75% of all known planets, and is therefore the most inclusive source of information on extrasolar planets. This detection method measures the drop in luminosity caused by a planet's transit. For example, a planet as small as the Earth transiting in front of a Sun-like star would cause a drop of about 100 ppm (parts per million) of the received starlight, which is just about current instrument capabilities, like TESS [30]. This means that planets smaller than the Earth or planets orbiting bigger stars would be difficult to detect with the Transit Photometry method due to current limitations.

Because of the geometry at play during transit, it has been shown that planets transiting closer to their host star or orbiting fainter stars are easier to detect [36]. This potentially hinders our understanding of the current exoplanet demographics around brighter stars or at longer orbital locations. That said, this bias due to technical limitations and geometrical constrains can be quantified using injection/recovery tests, which allows to correct subsequent estimations [17, 37].

It is generally true that long orbiting planets are harder to detect, because their orbital motion has effect on their host star that spans over larger time scales. Typically, planets with orbital period of more than two years happen to be detected either by direct imaging or gravitational microlensing. Note that planets detected through gravitational microlensing will never be probed again with that same method, making scientific reproducibility impossible in such cases [38].

As for the Radial-Velocity Spectroscopy method, planets are probed through the change in velocity of their host star projected along our line of sight. For example, the Spectrograph HARPS (High Accuracy Radial Velocity Planet Searcher) [32], in Chile, has a precision of about 1 m/s, which allows for the detection of Super-Earths or bigger planets (of course depending of the host star mass and the inclination angle). For comparison, the Earth orbiting around the Sun generates a Doppler shift of 9 cm/s, significantly below current instruments precision.

That said, it is worth mentioning that current instruments do not fail to meet their specifications because of the technology, but because of intrinsic stellar reasons that are discussed in the next subsection.

– Stellar Activity Noise Floor –

In spite of great technical resolution, the activity of host stars create additional noise in the light curves (for Transits) or Doppler records (for RV), which limit the actual resolution of the measurements. This noise cannot be avoided and sets a limit to the practical resolution of our technologies, hence the term "noise floor". The stellar activity noise floor originates from three phenomena: oscillations, granulation, and magnetic activity.

Stellar oscillations are the periodic contractions and expansions of the external layers of a star [39]. These oscillations usually occur during time scales of few minutes, and cause an additional noise of about 1 to 10 cm/s for Doppler measurements [40, 41].

Stellar granulation refers to the appearance of small-scale patterns at the surface of a star. These patterns are caused by convection in the outer layers with typical time scales ranging from few minutes to several hours. When integrated over the whole stellar disk, the various stellar granulations can account for additional noise on the order of 1 m/s [39, 42, 43].

Finally, stellar spots and plages are respectively dark and bright regions on the surface of a star caused by magnetic fields. These phenomena can lead to an additional noise varying from 40 to 140 cm/s for Doppler measurements [39, 44, 45].

For all the technical reasons mentioned before and stellar activity phenomena presented here, it is worth noting that the current population of surveyed exoplanet results from a very biased selection process. Only planets which have a significant contribution to the observed light of their host star can be detected. The sample of known exoplanets therefore includes mostly large and massive planets in close orbit, and smaller planets are typically found around fainer stars. Besides, it is yet not clear whether our own Solar System constitutes a typical stellar system, despite no Solar System equivalent discovered to date.

1.2 Artificial Neural Networks and Machine Learning: Towards Data Imputation Algorithms

Over the last decade, Data Science and Machine Learning has gained a lot of attention, and these tools are now applied in a lot of various domains. But what do we even mean when we talk about “Machine Learning”?

More broadly, Artificial Intelligence (AI) refers to programs or algorithms designed to mimic the human process of learning and making decisions. AI is a goal-oriented paradigm, which means that programs try to achieve predefined targets. Machine Learning (ML) is a framework of AI where a program learns and improves from experience. Learning is achieved by automatically finding patterns in existing data, without human intervention or hard-coded rules. Within the realm of ML, we find Artificial Neural Networks (ANNs) whose popularization has been made possible thanks to improvements in computational neuroscience. Because of their flexibility and high performances, ANNs have become the gold standard in ML. Deep Learning (DL) is the most recent form of ANNs, where the adjective “deep” refers to the number of layers within the network. As most ANNs tend to have at least few layers nowadays, the terminology for DL and ANNs are often used interchangeably.

The beginning of this Section presents a brief history of Artificial Neural Networks (Section 1.2.1) and goes over the fundamentals for training them (Section 1.2.2). Next, Machine Learning algorithms that are outside the scope of Artificial Neural Networks are being briefly introduced (Section 1.2.3). The remaining part (Section 1.2.4) delves into the specific challenges and typical statistical methods related to numerical data imputation, a problem comprehensively addressed by this thesis.

1.2.1 A Brief History of ANNs

Artificial Neural Networks (ANNs) were developed in an attempt to model the biological mechanisms of the brain. In 1943, McCulloch and Pitts proposed an algorithm to model biological networks of neurons [46]. Their model paved the way for further research on ANNs, not only to investigate the biological mechanisms of the brains, but also as a tool to enhance the capabilities of ML algorithms. Later in 1949, Hebb postulated that learning can be achieved because of neural plasticity [47]: “Cells that fire together, wire together”. Neuroplasticity is the ability to adapt connections between neurons with respect to sensory experiences. This is exactly how ANNs are trained nowadays (c.f. Section 1.2.2).

In 1958, the Perceptron was introduced by Rosenblatt [48]. The Perceptron is an ANN with one single layer of fully connected neurons, developed to solve binary classification tasks. However, the results were not convincing, and the research in AI was stagnating. In 1969, Minsky and Papert presented the limitations of the Perceptron [49] and computer processing power was still limited at the time. This state of affairs led to reluctant funding opportunities for research on ANNs: this was the beginning of the “AI winter”, a period of decreased interest for AI research.

Regarding the limited computational power of computers, and based on previous trends, Gordon Moore (CEO of Intel) postulated in 1975 that the number of transistors on microchips will continue doubling every two years. Famously known as Moore’s law, this trends has not been invalidated to this day, 50 years later.

The disillusion for ANNs lasted until computers were endowed enough computational power, in the 1990's. Meanwhile, Werbos proposed the backpropagation algorithm, which allowed to train ANNs [50]. In 1989, LeCun developed the first implementation of a Convolutional Neural Network (CNN) to recognize handwritten digits from the famous MNIST dataset [51].

Nowadays, most of our electronic devices are connected to the Internet, leaving a lot of information that are recorded and used to train ML models. Many famous datasets exist online, and yearly competitions attempt to push the performances of ML to their maximum, showing better-than-human performances on a variety of non-trivial tasks. AI tools are omnipresent in our society, allowing to avoid traffic congestion, correcting the spelling mistakes, delivering weather forecasts, or making meaningful recommendations on social media. Industries, companies, and public services also make advantage of ML tools: production chains and storage are optimized, medical diagnoses are guided, and self-driving robots are currently walking on Mars.

1.2.2 ANNs in Practice

Machine Learning tasks are usually split into three major branches: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is achieved by training on correctly labeled data (e.g. regression or classification tasks). Unsupervised learning tries to model distributions or find patterns in unlabeled data (e.g. clustering tasks or feature engineering). Finally, reinforcement learning consists in improving by trials and errors (e.g. playing games or walking robots).

– Various Flavours of ANNs –

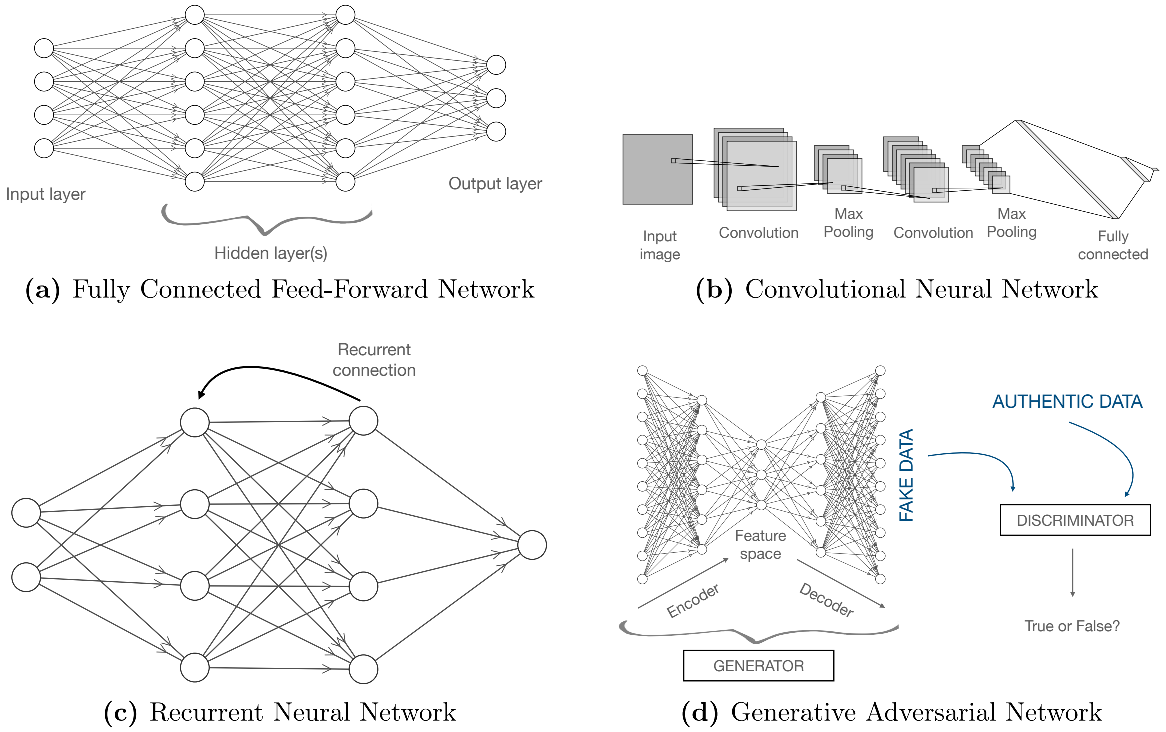
In order to tackle different tasks or to process different types of data structures, various architectures of ANNs have emerged over the recent years. The most standard design is the Fully Connected Neural Network (also referred as Multi-Layer Perceptron) [52] and consists of a concatenation of successive layers of neurons, where every neuron is connected to all its predecessors and successors from its neighbouring layers.

Introduced earlier, Convolutional Neural Networks (CNNs) [51] constitute another popular choice to work with spatially correlated data. Each layer is made of adaptive filters, and training a CNN consists in finding appropriate filters, whose role is to extract features from data. Typically applied to image data, the filters are 2-D small windows used to perform convolution over images. However, CNNs are not limited to the two-dimensional case. For example, 3-D filters can be used to work with videos [53] or real physical three-dimensional particles [54]. Besides, CNNs have also shown great performances in 1-D as well, typically when applied to sequential data like time series (more on that in Chapter 5) [55].

Most ANNs architectures are said to be feed-forward networks, because the information flows unidirectionally from the input layer (raw data) to the output layer (prediction of the network). But this does not have to be necessarily the case. For instance, Recurrent Neural Networks (RNNs) allow for loops in their architecture. As its name suggests, the Long Short-Term Memory (LSTM) is a type of RNN that holds both a short-term and a long-term representation of the data while being sequentially processed [56]. RNNs are well suited to work with sequential data, like for text processing or speech recognition tasks.

That said, most RNNs have become obsolete with the coming of Transformer models, which rely on the multi-head attention mechanisms [57]. Because of their recurrent architecture waiting to process the next input, RNNs cannot be trained in parallel, which makes them very time-consuming and during training. Instead, the attention mechanism allows the Transformer to adaptively select what part of the input sequence is deemed relevant for a given task, now enabling to provide the whole data at once, and not in a sequential manner anymore. Transformer models are nowadays extensively used for Natural Language Processing tasks, but they can also be applied to various other domains like Computer Vision [58], Speech Recognition [59], or even Symbolic Regression (more on that in Chapter 6).

Generative models encompass many architectures, and refer to models designed to learn the probability distribution of a dataset in order to draw new samples from it. One famous example is the Generative Adversarial Network (GAN) [60], which pits a Generator and a Discriminator against each other. The Generator outputs data that resembles data in the original dataset, while the Discriminator should assess whether presented samples are genuine (real data) or fake (generated by the Generator). After convergence, the Generator has learned the distribution of the original dataset.



(a) Fully Connected Feed-Forward Network

(b) Convolutional Neural Network

(c) Recurrent Neural Network

(d) Generative Adversarial Network

Figure 1.5: Architectures of frequently encountered ANNs

Figure 1.5 shows the structure of few common ANN models mentioned in this section. There exist virtually infinitely many ways to arrange the connections between artificial neurons within a network, and each architecture makes specific assumptions about the data being processed. For example, Graph Neural Networks (GNNs) can be used to process data that can be represented as graphs [61] (more on that in Appendix A).

– How to Train Them? –

Connections between neurons are linear, which allows to efficiently compute a forward pass by use of matrix multiplications and offers the possibility to parallel the training when multiple GPUs are available. However, non-linear functions should be used in between layers to break the linearity of the network, otherwise the entire ANN would boil down to a single Perceptron. These intermediary non-linear functions are called activation functions, and the most common is the Rectified Linear Unit (ReLU) activation function, as defined by $f(x) = \max(x; 0)$. In spite of its simplicity, the ReLU activation function allows for the training of complex ANN architectures [62]. There exist other standard activation functions used for specific tasks, like $f(x) = \tanh(x)$ for an output in the range $[-1, +1]$, or the softmax activation function to convert any list of real number into probabilities (typically for classification tasks).

Training ANNs involves minimizing a loss function – or maximizing an objective function depending on the task at hand. Oftentimes, the loss function is chosen to be the Root Mean Square Error (RMSE) for regression tasks, or the categorical cross-entropy for classification tasks, respectively defined as $\varepsilon_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}$ and $\varepsilon_{\text{CE}} = -\sum_{i=1}^{N} \widehat{y_i} \log(y_i)$. More complex architectures use other types of loss function, e.g. the Generator of a GAN is trained by maximizing the cross-entropy of the Discriminator in a zero-sum game manner, or the triplet-loss to maximize the difference between two categories of data with respect to an anchor [63].

The process of training an ANN is the process of finding optimal parameters (weights and biases for the connections between neurons) to minimize the loss function computed over the training dataset. However, due to the usually large number of parameters in the ANN, finding the global minimum of the loss function is practically impossible because of the curse of dimensionality imposed by the huge parameter space [64].

In practice, Stochastic Gradient Descent algorithms are used: the entire training dataset is split into smaller batches, over which the loss function is computed. Given a loss function $\mathcal{L}(\theta)$ where θ is the vector of parameters, one update using Stochastic Gradient Descend consists in computing the gradient $\nabla_\theta \mathcal{L}(\theta)$ with respect to θ over a mini-batch and updating the values of θ according to:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$$

where η is the step size of the gradient descent (often referred as "learning rate" in ML). Once the entire dataset has been processed by mini-batches, it is shuffled and the process is done again. This corresponds to one epoch.

Improvements over the traditional stochastic gradient descent algorithm have been proposed, especially for the automatic scaling of the learning rate with respect to the loss function gradients. The Adam optimizer [65] (for Adaptive Moment Estimation) seems to be the state-of-the-art algorithm, and it has become the standard practice for ML research and development.

1.2.3 Machine Learning outside ANNs

As presented in the last two Sections, Artificial Neural Networks are brain-inspired Machine Learning models which use large number of neurons and weights. But Machine Learning models are not restricted to ANNs, and this Section introduces traditional Machine Learning algorithms outside the scope of ANN. Most of the Machine Learning methods presented in this Section will be useful for the next Section, when talking about Numerical Data Imputation algorithms (Section 1.2.4).

– Linear Regression –

Linear Regression [66] is certainly the oldest and most established Machine Learning algorithm, where a set of predictors X are used to model the value of a response variable Y through the linear relationship: $Y = AX + b$. Despite its simplicity, linear models are still extensively used nowadays, as they provide with a closed form solution and allow for a simple interpretation.

– Decision Trees and Random Forests –

A Decision Tree is a hierarchical model with a tree-like structure that can be used for regression or classification tasks. Decision trees iteratively split the input space into regions using a selected feature, eventually leading to a partition [67]. Predictions are obtained by aggregating the response variable values for the observations in the same leaf node as the input.

Because Decision Trees are known for overfitting the training sample, Random Forest algorithms propose to aggregate one more time the predictions of many Decision Trees in order to obtain better predictive power [68]. Random Forests can be computationally expensive, but lead to great performances on numerical datasets.

– k-Nearest Neighbors –

Similar to Decision Trees, the k-Nearest Neighbors algorithm (k-NN) can be used both for classification and regression tasks. Given an input observation, the k-NN algorithm searches for the k closest training example, and uses an aggregate (either mean or majority voting) of the selected k observations to estimate the value for the input point [69].

This algorithm is appreciated for its simplicity and its interpretability. The hyperparameter k corresponds to the number of neighbors to be selected, and needs to be fine-tuned. Small values for k tend to lead to high variability in prediction with smaller bias, and higher values for k tend to decrease the variability but increase the bias.

1.2.4 Numerical Data Imputation Algorithms

A major focus of this PhD thesis is to understand the challenges related to numerical data imputation, to analyze and compare the performances of various data imputation algorithms and to propose a new strategy to impute numerical missing data (see Chapters 2 and 3). This Section provides an introduction to the subfield of numerical data imputation and presents traditional strategies.

– What is Numerical Data Imputation? –

In practice, datasets are never clean and perfect. Missing data is a prevalent problem in ML, and if not handled correctly can even lead to biased or potentially wrong conclusions. Data might be missing if lost, degraded, censured, or simply not measured because of practical reasons. Certain ML algorithms can also not be used with incomplete datasets: for example, the standard Principal Component Analysis (PCA) [70] can only be applied to completely observed datasets.

Data imputation algorithms aim to address this problem by estimating missing values, and a wide range of tools have been developed over the years (see below). One of the most popular application of data imputation algorithms consists of recovering missing parts of an image, also referred to as in-painting. For image recovery tasks, Deep Learning models have shown promising results and have therefore become the standard solution [71].

However, it is worth noting that typical features for images greatly differ from typical features for numerical datasets. Most specifically, images are said to be "sparse" data, because they can be compressed by a large factor without suffering from an important loss in data. For example, an image of size $256 \times 256 = 65,536$ pixels can be condensed into few features after being processed by a CNN. Assuming we have 64 features to describe the original image, this corresponds to a compression by a factor of approximately $1,000$. On the other hand, let us consider a typical tabular dataset with $1,000$ rows and 10 columns and comprising of numerical values. In the best scenario, this numerical data (in practice, a matrix) might be reduced to a matrix of size $1,000 \times 2$ after PCA, which corresponds to a compression of only a factor 5, despite potentially a great loss in information. For that reason, there is no guarantee that Deep Learning models remain the best solutions for the statistical analysis of smaller tabular numerical datasets.

This PhD addresses the problem of data imputation for tabular numerical datasets, i.e. datasets with numerical values which we can arrange as a matrix, with rows (observations) and columns (variables/features).

Let's denote $x \in R^D$ the (unobserved) ground-truth for an observation in dimension $D > 2$, and $m \in 0, 1^D$ its corresponding missing mask. The effectively observed data is presented as $\tilde{x} = x \odot m$, where $\odot$ denotes the element-wise multiplication. The goal is now to retrieve x from $\tilde{x}$. The probability distribution of the missing mask, $p(m)$, is called the missing data mechanism, and varies accordingly to missing data scenarios. Following the standard classification of Little and Rubin [72], there are three scenarios to account for missingness in a numerical dataset: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). In MCAR, we assume that the missing data mechanism is independent from the data, such that $p(m|x) = p(m)$. In MAR, we assume that the observed data can fully explain the reason why data is missing, and we can write $p(m|x) = p(m|\tilde{x})$. Finally, the MCAR scenario encompasses everything else, and the reason why data is missing might depend on the missing data themselves.

– Standard Numerical Data Imputation Algorithms –

Historically, data practitioners confronted to missing value problems with numerical records used to delete entire observations in order to obtain a completely observed matrix of data and easily conduct subsequent analysis. This "strategy", called list-wise deletion, is of course the worst course of action to be possibly undertaken, as it degrades the statistical power of tests, can lead to biased samples, and it merely a waste of potentially useful information [73].

Instead, we can try to estimate missing values. A once-common strategy of imputation is the hot-deck, referring to historical punched cards of data being currently processed, hence "hot". This consists in using the last observed value as an estimate for another missing value. Although better than list-wise deletion, this imputation strategy remains crude and has been shown to bias conclusions, e.g. exaggerating the effectiveness of drugs in tests [74].

A more sophisticated strategy consists in imputing a missing value for a given feature by the mean or the median of that feature computed across all other observed values. This method preserves the mean of that feature but decreases the correlation with other features. Intuitively, we can see that all missing values will all be imputed using the same numerical value, regardless of other observed properties. To address this drawback, several means can be computed over different classes, but this imputation strategy always leads to a decrease in correlation, and therefore biased subsequent multivariate analyses [75].

More recent statistical methods allow for greater flexibility in the way missing values are estimated. The kNN-Imputer [76] computes pairwise distances between of observations, selects the k closest neighbours and use their mean in order to estimate missing values. The kNN-Imputer algorithm has been shown to provide more accurate and robust missing value estimates than using the feature mean.

There also exist multiple imputation strategies, seeking convergence towards on optimally imputed dataset. Multiple Imputation Chained Equations (MICE) [77] refers to an iterative imputation algorithm: the missing values are first filled with initial guesses (typically the column mean), and the missing values of each column are imputed one at a time using the other columns as predictors. The algorithm repeats for a fixed number of loops through all columns in the original dataset, or until convergence is attained following a user-defined criterion. The traditional version of MICE uses linear regressions to predict the missing values. But other more sophisticated version of the MICE algorithm can leverage more flexible algorithms like MissForest which makes use of Random Forests [78].

Finally, Deep Learning based data imputation algorithms have been recently developed with the hope to provide better imputation results than standard methods. Most notably, GAIN is a Generative Adversarial Network model tailored for numerical data imputation [79]. GAIN aims at filling missing values from numerical datasets by generating fake data that cannot be discriminated from other real observed data. Besides, there exist frameworks to fit Variational Auto-Encoders (VAE) to incomplete datasets, such that it become possible to use a latent distribution learned by the VEA to impute missing values [80]. That said, and in spite of impressive results for Deep Learning methods when dealing with text, images, or videos, it is still unsure whether these approaches are to be preferred for numerical datasets, and simple traditional imputation methods appear to yield best results [81]. In particular, it has been shown that tree-based methods still remain state-of-the-art for tabular numerical data [82].

– Challenges in Numerical Data Imputation –

As data imputation often tends to be overlooked, taken for granted, or not considered a problem, we will see that this is instead a very complicated task which determines the quality and reliability of subsequent analyses. The main challenge is to impute values while considering as many patterns from within the observed data as possible, but also without creating artifacts. This task is particularly complex in the presence of multimodal datasets.

Naturally, data practitioners are interested in retrieving a complete dataset from an incomplete one. As a result, most data imputation methods return a single point estimate, therefore having to choose a statistical quantity for imputation (typically the mean or median). But this choice is arbitrary, and does not capture the real univariate data distribution, which can be skewed, bimodal, show clusters, or having long tails. Also, imputing one column at a time (e.g. MICE methods or the kNN-Imputer) do not allow to consider multivariate dependencies when estimating missing values. Chapter 2 of this PhD thesis will cover that part.

In addition, we are also interested in the computational cost of the chosen data imputation method. Regardless of how important data imputation can be, estimating missing values is never a final step and further analysis is always performed. Therefore, one cannot (or does not want to) spend too much time and efforts into optimizing the data imputation procedure.

Ultimately, this PhD work aims at learning an interpretable Generative Model from an incomplete tabular dataset of numerical values, and while preserving the structure in the original dataset. The results using the newly proposed algorithm of this work are presented with the NASA Exoplanet Archive in Chapter 4.

1.3 What this PhD Thesis will cover

The theme of this PhD thesis revolves around Machine Learning tools for the characterization of planetary systems. The major focus of this work intents to thoroughly compare numerical data imputation methods, develop a new numerical imputation algorithm (Chapters 2 and 3), and apply it to the NASA Exoplanet Archive (Chapter 4). Additional complementary projects involve the development of various Artificial Neural Networks for (exo)planetary research, and its application towards scientific discovery (Chapters 5 and 6, and Appendix A).

1.3.1 Making Full Use of the NASA Exoplanet Archive

The initial motivation for this PhD work is the leverage the underutilized NASA Exoplanet Archive. The NASA Exoplanet Archive compiles information from tens of thousands of studies and gather data for all confirmed exoplanets into an exhaustive database. Although everyone can freely access this dataset and download it, the NASA Exoplanet Archive remains rarely used, and few work aim at analyzing the demographics and trends within the global exoplanet population.

It is worth noting that the NASA Exoplanet Archive does present a lot of missing values. As presented in Section 1, every discovery method has its own advantages and drawbacks, oftentimes leading to physically unobservable properties. For example, the Radial Velocity (RV) method only provides a minimum mass and no information on the planet radius, while the Transit method provides the radius but no mass information. As such, the NASA Exoplanet Archive presents a lot of missing values and appears as an interesting database to try numerical data imputation methods (c.f. Chapter 4).

Besides, the exoplanet population does show complex dependencies and multimodal distributions. For instance, there is no clear boundary between a Super Earth and a Neptune-sized gas planets, both of them potentially having similar radii but very different masses. Moreover, statistical tools are required to utilize the information of the minimum mass returned by RV methods, acting like censored data. As a result, data imputation is not straightforward for the NASA Exoplanet Archive, and new tools capable of capturing multimodal or complex distributions have to be developed for this purpose (see Chapter 3).

In the long run, this work aims at helping the identification of interesting but partially observed planets, and help better characterizing them. Typically, this work could assist in selecting future missions' targets, especially small rocky planets which size make it hard to observe and are often poorly characterized for that reason. However, small terrestrial planets orbiting in the habitable zone of their host star are of particular interest in the search for extraterrestrial life biomarkers.

1.3.2 A New Numerical Data Imputation Method for Multimodal Datasets

The NASA Exoplanet Archive has been the perfect opportunity for me to delve into numerical data imputation methods and better grasp the current challenges of this field. Over the course of my PhD, I quickly realized that current imputation methods do not allow for enough flexibility in estimating missing values as they often provide a single point estimate so to make the dataset ready for further analysis.

Inspired by the simplicity and flexibility of the kNN-Imputer, I decided to develop a new numerical imputation method combining both the kNN-Imputer framework and kernel methods in order to obtain a probability distribution for imputation, rather than a single point estimate. This thesis presents the properties of the numerical data imputation algorithm I propose, called the kNN×KDE. I show when the proposed imputation method works best in the presence of multimodal distributions, and how to use it in practice for interested data practitioners.

1.3.3 Application of ANNs to Planetary Systems

Finally, I dedicate some of my time as a PhD student to collaborate with astrophysics researchers and develop new Machine Learning tools for planetary research. Because they fit the same theme, these works will be included in this PhD thesis. They include three Machine Learning algorithms, each employing a different Neural Network architecture, therefore making it a great opportunity for me to learn new ANN methods (see Chapters 2 and 3).

The first one is a Convolutional Neural Network (CNN) used for time-series analysis of the orbital element of three-body systems. If the orbital mechanics for two-body systems has been long studied, we also know that three-body (or more) systems exhibit a chaotic behaviour. It is therefore very complicated, if not impossible, to predict their fate in the far future. With Alessandro Alberto Trani, we have developed a CNN for the analysis of the Keplerian orbital element of three-body systems, seen as multiple stacked time-series. Our best model is capable of predicting the stability of such chaotic systems with over 95 % accuracy (c.f. Chapter 5)

Next, and under the supervision of Yasushi Suto, we have been working on the conditions of the discovery of Neptune in 1846 (see Appendix A). For this project, I developed a Graph Neural Network (GNN) which allows to model complex dependencies between objects, which can also have arbitrary features. In this case, objects are planets (and the Sun), their feature is their mass, and the relationship between these objects is their pair-wise gravitational influence. We employ the GNN to estimate the gravitational relationship between planets, and try to retrieve the Universal Law of Gravitation of Newton, completely empirically and without the rules of calculus [83]. This project aims at quantifying up to what extend the exact analytical law of Newton was necessary to allow for the discovery of Neptune by looking at Uranus' irregularities [84]. In other words, assuming we did not have calculus 200 years ago, but unreasonably advanced computers and sensors instead, would we be able to automatically discover Neptune using AI?

Last, I performed an internship at OMRON SINIC X (OSX) over the summer 2023, during which I learned the technology behind Transformer models [57] and the challenges of Symbolic Regression [85]. During this internship, I developed a Transformer model tailored for Symbolic Regression in the context of automatic/assisted scientific discovery (c.f. Chapter 6). Symbolic Regression is a type of regression where both the skeleton of the equation and its constants have to be estimated. This is a very complicated problem because the space of possible mathematical expressions to search from grows exponentially with the number of functions allowed to be combined. Even though not directly related to astrophysics, this project aims at direct applications of AI/ML to scientific fields, and I found that most difficulties faced in this project were similar to the difficulties I faced with the Neptune project.

1.4 Outline

Chapter 2: Numerical Data Imputation with Deep Learning does not work well. This chapter presents a brief comparison for numerical data imputation with various missing rates and missing data scenarios involving two recent Deep-Learning methods using GAN framework: GAIN [79] and MisGAN [86]. I compare with the standard kNN-Imputer as benchmark, and show that the kNN-Imputer remains better than the newly proposed GAN Deep-Learning models.

Chapter 3: The kNN×KDE. Following the superiority of the kNN-Imputer for numerical data imputation, I developed the kNN×KDE to leverage the performances of the kNN-Imputer and the density estimation of kernel density methods. Besides, this study evaluates and benchmarks with more algorithms than the previous chapter, and also uses more datasets. The published article calls for more consideration during data imputation, especially when unambiguous imputation is not possible.

Chapter 4: Imputation of the NASA Exoplanet Archive using the kNN×KDE. After introducing the kNN×KDE in the previous chapter, this chapter presents its application to a database of particular interest: the NASA Exoplanet Archive. Imputing missing values in the Exoplanet Archive using the kNN×KDE allows to analyze the various probability distributions for missing properties, in particular planet masses and radii. Bimodal, trimodal, skewed, or heavy-tailed distributions revealed interesting facts about the underlying multi-dimensional planet demographics. We also used the kNN×KDE as a generative tool to simulate new artificial planets and better investigate the demographics of the known population of exoplanets.

Chapter 5: Hierarchical Triple Systems Stability with CNNs. This chapter presents a newly developed CNN model to predict the long-term stability of hierarchical triple systems, known for exhibiting chaotic behaviour. The CNN is presented several time series corresponding to the evolution of the Keplerian elements of the system, and performs a binary classification task between stable and unstable systems. Our best model has an area under the ROC curve of more than 95 %, and allows to bypass the complete integration of the system, providing a speed-up by a factor of 200.

Chapter 6: Symbolic regression with Transformer Models. The final chapter of my thesis presents the challenging work I performed during my internship at OMRON SINIC X. I have developed a Transformer Model tailored for Symbolic Regression. Once trained, the models takes a matrix of numerical values as input, and outputs a tentative mathematical expression to describe the provided data.

Appendix A: Automatic Rediscovery of Neptune without calculus. This project has not led to a published article yet. In this appendix chapter, I briefly present the methodology which consists in using a Graph Neural Network (GNN) to model the relationships between the bodies in our Solar System. I trained the GNN using simulated data of the planets motions up to Uranus. The aim is to experimentally retrieve the Universal Law of Gravitation of Newton as accurately as possible. Then, I employ the GNN-learned law of gravitation to evolve the bodies of the Solar System once again, in the hope to find discrepancy between the "real" data (with Neptune) and the simulation of the GNN. A tentative conclusion of this work is that the exact analytical Law of Gravitation was key to allow for the discovery of Neptune in 1846.

Chapter 2

Numerical Data Imputation with Deep Learning does not work well

As my PhD work originally focused on estimating numerical missing values, the newly proposed Generative Adversarial Imputation Nets (GAIN) [79] seemed to be a promising method. This Chapter presents empirical results and benchmarks for tabular numerical data imputation using Deep-Learning GAN models.

2.1 Context

GAIN is a Generative Adversarial Network (GAN) tailored for numerical data imputation. Particularly interested by the potential of GAIN for numerical data imputation, I decided to experiment this method using both real-world and simulated numerical datasets.

Besides, MisGAN is another GAN model aiming at data imputation tasks [86]. Unlike GAIN, MisGAN was originally developed for image imputation tasks, also known as inpainting. However, its framework can be easily adapted to the case of tabular numerical data.

Quickly enough, I came to the realization that not only GAIN and MisGAN performed poorly at recovering simple datasets missing values, but I was also unable to reproduce the results published by the authors of GAIN, even when following the methodology provided in their original article [79]. Moreover, training generative models on datasets presenting missing data has been shown to be a particularly challenging task, even without using deep-learning methods [87].

Interestingly, the traditional kNN-Imputer appeared to provide reasonable imputed values in a much smaller inference time. These results were in accordance with existing numerical data imputation benchmarks [81, 88–91] advocating that the kNN-Imputer and MissForest, in spite of being simple algorithms, actually perform best for numerical imputation.

In light of these impromptu findings, I decided to systematically compare the kNN-Imputer against GAIN and MisGAN, using a broader range of datasets and following controlled experiments in various missing data scenarios (MCAR, MAR, and MNAR) and with various missing rates (from 10% to 80%, with steps of 10%).

Because of computational time constraints, the hyper-parameter for the three data imputation algorithms were optimized on the well-behaved synthetic dataset (mixture of Gaussians), and later adaptivately scaled on other datasets depending on their number of observations. The scaling of the hyper-parameter has clear limitations, and extensive hyper-parameter tuning has been performed in the next Chapter, when more data imputation methods were considered (see Section 3).

The methodology, the experiments, the datasets, and the code can be accessed on GitHub[1].

2.2 Published article

Lalande Florian and Doya Kenji. *Numerical Data Imputation: Choose kNN Over Deep Learning.* Similarity Search and Applications, Lecture Notes in Computer Science, Volume 13590, Springer, 2022.

I presented these results in an oral presentation and a poster session at the Similarity Search and Application (SISAP) 2022 Conference (Bologna, Italy) on October 6, 2022.

2.3 Conclusion

The results presented in the above-mentioned article indicate that the kNN-Imputer performs better than GAIN, while requiring a single hyper-parameter to be tuned, its number of neighbours k. Although the authors of MisGAN explain that their framework can be extended to tabular numerical datasets, I found that in practice MisGAN performs disastrously for numerical data imputation.

A tentative explanation is that GAIN and MisGAN, like all Generative Adversarial Networks, are difficult to optimize because of training instabilities, mode collapse problems, potential impossibility to converge, or not well defined loss function [92]. In practice, I indeed realized that GAIN strongly suffers from mode collapse problems, and systematically returns identical estimates for all missing value in a column, somehow similar to the column-mean imputation strategy, except that the imputed value is not necessarily the column mean.

It is worth mentioning that some other "deep-learning" approaches do exist, which are not GAN methods. In particular, Variational Auto-Encoders (VAEs) can use deep latent variables to model the missing data mechanism itself [93, 94].

Besides, I also noted that the kNN-Imputer is much faster to provide inference results for reasonable sized datasets. This is however not true anymore for large datasets, as nearest-neighbours approaches require pairwise distances to be computed and suffer from the curse of dimensionality. For large enough datasets, GAIN might have an advantage over the kNN-Imputer.

Finally, and on a more personal note, this preliminary work for my PhD has been the opportunity to realize that Artificial Neural Networks (ANNs) often do not outperform long-established statistical methods for tabular numerical datasets. As Grinsztajn et al. point out in their study *Why do Tree-based Models still outperform Deep Learning on Tabular Data* (2022) [82], ANNs are not robust to uninformative features, cannot

[1] https://github.com/DeltaFloflo/imputation_comparison

learn very irregular functions, and favour overly smoothed functions, which prevent them from good results on tabular and numerical datasets. Also, I remain puzzled about the imputation results proclaimed by GAIN. Even after contacting the authors of GAIN, it is still impossible for me to reproduce their results. In light of this, I invite you to have a look at the following CrossValidated post[2], as I am still hoping for a satisfactory answer.

[2] https://stats.stackexchange.com/questions/612554/numerical-data-imputation-generative-adversarial-imputation-nets-gain-not-rep

Chapter 3

The kNN×KDE

Following my personal disillusionment with GAN methods for numerical data imputation (see Chapter 2 and Reference [95]), I decided to develop a new tool for the imputation of numerical datasets. More specifically, I wanted a numerical data imputation algorithm that returns a probability distribution to capture the multidimensional relationships.

3.1 Context

Most tabular data imputation algorithms return a single point estimate, often taken as the mean (or the median) over few samples. While computing the mean allows to obtain more robust estimates for the missing values, this reduces the available information from a rich distribution into a single point, potentially neglecting important features (e.g. several modes or skewness).

To illustrate this problem, I focused on three simulated datasets with simple structure (see Figure 3.1) and designed an imputation strategy capable of capturing the conditional probability distribution of the missing values given the observed values, instead of returning a point estimate. The chosen toy datasets illustrate how most traditional imputation methods work well in the unambiguous case with a unique solution for the missing value, but fail when several possible values are consistent with the original data structure.

The proposed imputation strategy combines the flexibility of the kNN-Imputer [76] and the simplicity of Kernel Density Estimation (KDE) [96, 97], hence its name: the kNN×KDE. Note that previous attempts of generalizing KDE methods to incomplete datasets have been proposed and offer a comprehensive theoretical analysis [98, 99], but remain computationally too expensive for practical purposes. Instead, the kNN×KDE offers a computationally cheap alternative, simple to implement and use in practice, but does not offer new theoretical insight.

For a given missing pattern, the kNN×KDE computes pairwise distances between the observations to be imputed and all potential donors. The metric used to compute pairwise distances is adapted from the `NaN-Euclidean-distance` [100] used by the kNN-Imputer, but does not overlook pairs of features where at least one observation is missing. The new distance, called `NaN-std-Euclidean-distance`, instead uses the standard deviation of the column when a value is missing and the distance cannot be computed. The distance

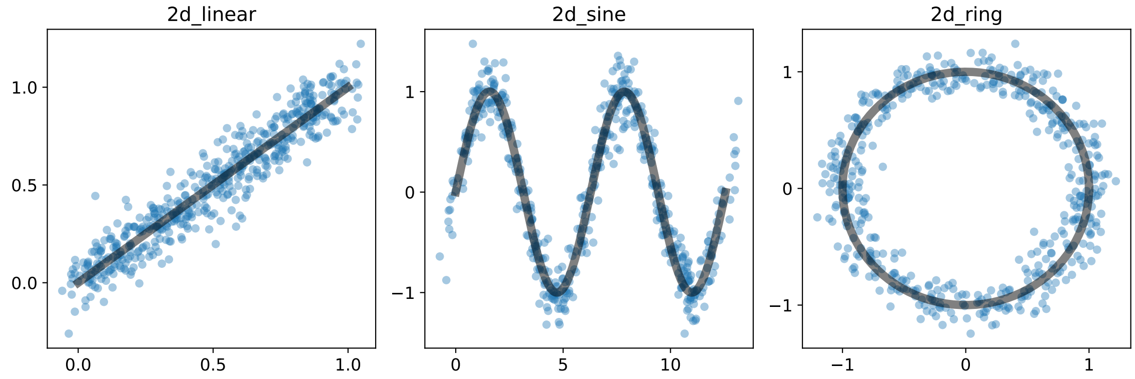


Figure 3.1: Three basic synthetic datasets with $N = 500$ observations. `2d_linear` is a bijection, `2d_sine` is a surjection, and `2d_ring` displays a ring and is therefore not a function in the euclidean space.

d_{ij} between observations i and j is computed as:

$$d_{ij} = \sqrt{\sum_{k \in \mathcal{D}_{\text{osb}}} (x_{ik} - x_{jk})^2 + \sum_{k \in \mathcal{D}_{\text{miss}}} \sigma_k^2}$$

where $\mathcal{D}_{\text{obs}}$ is the set of indices for commonly observed features in observations i and j, $\mathcal{D}_{\text{miss}}$ is the set of indices for features where at least one observation i or j is missing (i.e. the complementary set of $\mathcal{D}_{\text{obs}}$), and σ_k is the standard deviation of feature k.

Next, potential donors are weighted through a softmax function, such that these weights can be interpreted as probabilities of being selected for imputation. The softmax function takes a hyperparameter τ, the softmax temperature, enabling to treat the potential neighbours in a continuous way instead of the discrete version the traditional kNN algorithm. Samples are then drawn N_{draws} times, and gaussian noise with variance corresponding to the kernel bandwidth h is added. The pseudo-algorithm of the kNN×KDE is given in Figure 3.2.

3.2 Published article

Lalande Florian and Doya Kenji. *Numerical Data Imputation for Multimodal Data Sets: A Probabilistic Nearest-Neighbor Kernel Density Approach.* Transactions on Machine Learning Research (Reproducibility Certification), 2023. ISSN 2835-8856.

3.3 Conclusion

The imputation performances of the kNN×KDE have been evaluated using the RMSE and the log-likelihood score. On the one hand, the RMSE is computed between the ground truth and the imputed value, which allows to assess how close the estimate is from the (unknown) true value in the case of missing data. The RMSE score is commonly

```
Hyper-parameters: Softmax temperature $\tau$; Kernel bandwidth $h$; Nb draws $N_{\text{draws}}$

Data: Incomplete numerical data set $X$
min-max normalization in the interval $[0, 1]$;
for each missing pattern do
    $X_{\text{imp}} \leftarrow$ data_to_impute $(X, \text{missing pattern})$;
    $X_{\text{don}} \leftarrow$ potential_donors $(X, \text{missing pattern})$;
    $d_{ij} \leftarrow$ NaN_std_Euclidean_Distance $(X_{\text{imp}}, X_{\text{don}})$;
    $p_{ij} \leftarrow$ softmax $(-d_{ij}/\tau)$;
    for each row in $X_{\text{imp}}$ do
        $r \leftarrow$ sample $N_{\text{draws}}$ rows from $X_{\text{don}}$ with probabilities $p_{ij}$;
        $e \leftarrow$ sample noise $N_{\text{draws}}$ times from $e \sim \mathcal{N}(0, h)$ with dimension $K$;
        imputation_samples $\leftarrow X_{\text{don}}[r] + e$;
    end
end
min-max denormalization;
Return: imputations_samples
```

Figure 3.2: Pseudo-code for the kNN×KDE. The kNN×KDE has three hyperparameters: the softmax temperature τ controls the tightness of the neighbourhood, the shared kernel bandwidth h is used for the Gaussian kernels, and the number of returns samples N_{draws} determines the resolution of the probability distribution estimates. By working with missing patterns (c.f. main loop), the kNN×KDE only selects neighbours which will preserve the original structure of the dataset, unlike other traditional numerical methods that work one column at a time.

used to assess numerical data imputation algorithms performances. On the other hand, the log-likelihood score is a newly proposed metric for data imputation, and enables to assess whether the (unknown) ground truth falls within the expected probably distribution returned by the chosen data imputation methods. For each missing cell, we adapt the kNN-Imputer, MissForest, MICE, and the Column Mean imputation method to return the mean and the standard deviation, and use a Gaussian probability distribution to compute the likelihood of each missing value for each imputation algorithm.

In addition to the three simple synthetic datasets, 12 real world datasets have been curated for evaluation. The imputation performances are evaluated with increasing missing rates (from 10% to 60%) and in various missing data scenario (MCAR, MAR, MNAR). Eight numerical data imputation methods are compared:

- The newly proposed kNN×KDE [101]
- The traditional kNN-Imputer [76]
- MissForest, iterative imputation algorithm using Random Forests [78]
- MICE (Multiple Imputation Chained Equations) [77]
- SoftImpute, a matrix completion algorithm [102]
- GAIN, Generative Adversarial Imputation Nets [79]
- Column Mean (benchmark 1)

- Column Median (benchmark 2)

Results using the RMSE indicate that those eight numerical data imputation methods can be divided into 2 tiers: the kNN×KDE, the kNN-Imputer, MissForest, and MICE always outperform GAIN, SoftImpute, the column Mean and the column Median. As for the log-likelihood scores, the kNN×KDE and the kNN-Imputer respectively provide best and second performances.

The kNN×KDE was particularly designed to provide meaningful probability distributions for imputation, and was therefore expected to perform well on the log-likelihood score metric. But it also surprisingly outperformed (although by a small margin) other traditional imputation algorithms when looking at the RMSE results.

An online repository[1] provides all algorithms and data used in this work. In addition, I created easy to use Jupiter Notebooks to test and reproduce the results presented here. This work has received the Reproducibility Certification from the TMLR (Transaction on Machine Learning Research) Journal.

[1] https://github.com/DeltaFloflo/knnxkde

Chapter 4

Imputation of the NASA Exoplanet Archive using the kNN×KDE

Following the development of the kNN×KDE (see Chapter 3), the next step consists in applying this new tool to the NASA Exoplanet Archive. Instead of a blunt point estimate of the missing values, the kNN×KDE provides probability density estimates which allow further analysis to characterize the unobserved properties of exoplanets. The work presented here was done under the close supervision of Elizabeth Tasker, associate professor at the Japan Aerospace Exploration Agency, Institute of Space and Astronautical Science (JAXA, ISAS).

4.1 Context

The NASA Exoplanet Archive neatly compiles thousands of studies to provide with a convenient table listing all confirmed exoplanets to date. This is a living database and is updated on a weekly basis.

As mentioned in introduction of this PhD thesis (Chapter 1), the diversity of exoplanet detection methods implies various patterns of missing data. For example, the transit photometry method provides only the radius of detected exoplanets, and the radial velocity method provides a minimum mass measurement, which does not have to be close to the actual planet mass. Because of its missing values, all-embracing studies and demographics analysis remain challenging. Besides, the NASA Exoplanet Archive itself is not often studied and is instead seen as a living repository including exhaustive information on the population of known exoplanets.

Data imputation results are split into two scenarios. The transit scenario aims at concealing and retrieving the observed masses in the archive. The radial velocity scenario consists in concealing both the mass and radius and simulating minimum mass measurements. We then perform a convolution with the probability distribution given a minimum mass measurement, resulting in a new probability distribution taking into account both information from the other planets in the archive and the minimum mass measurement.

Building up on the work of Tasker, Laneuville, and Guttenberg [16] (hereafter TLG2020), the aim of this study is to estimate missing planets masses and radii while being able to use the whole exoplanet archive. Indeed, as TLG2020 proposed a modified Boltzmann

Machine (mBM), they needed to restrict their training dataset to completely observed planets (i.e. no missing value possible during training). They chose to focus on six planets properties: planet radius, planet mass, planet orbital period, planet equilibrium temperature, host star mass, and number of known planets in the system. With these six parameters, their training dataset was reduced to 550 planets.

With the kNN×KDE, not only were we able to use the whole exoplanet archive – bringing the dataset from 550 to 5,251 planets as of February 2023 – but also could we decide to add more properties (like the planet orbital eccentricity or the host star metallicity) and assess their impact on the estimated planet masses and radii without having to compromise on the dataset size. In addition, the probability densities provided by the kNN×KDE allow to leverage the information of the minimum mass available for radial velocity detections, by performing a convolution (which would be impossible with standard numerical imputation algorithms as they just return a point estimate).

The work presented in this chapter first compares several traditional data imputation algorithms performances against the kNN×KDE and the mBM of TLG2020. We then analyze several interesting probability distributions for retrieved or unknown masses and radii of planets. Lastly, we investigate the behaviour of the kNN×KDE as a generative model and perform unsupervised learning on the now complete exoplanet dataset.

The rest of this chapter presents the draft for an article submitted to the Astronomical Journal, but not published yet. The following of this chapter has been written in collaboration with Elizabeth Tasker.

4.2 Background and Related Work

Since the first discoveries in the early 1990s, over 5,500 planets have been discovered outside our Solar System. While the planets orbiting our Sun can be categorized as either rocky or gaseous simply depending on their orbital period, the myriad of sizes and orbits of the planets detected around other stars point to a multitude of formation pathways that are influenced by a wide range of environmental factors.

Dedicated survey missions such as Convection, Rotation et Transits planétaires (CoRoT), the Kepler space telescope, and the Transiting Exoplanet Survey Satellite (TESS), alongside ground-based search instruments and programs that include the High Accuracy Radial Velocity Planet Searcher (HARPS), Wide Angle Search for Planets (WASP) and Optical Gravitational Lensing Experiment (OGLE) are trying to build a census of planet types. This has resulted in the construction of a large archive of data for the properties of the discovered planets. Such an archive is an invaluable resource for identifying patterns and trends that can uncover the dominant factors that determine planet evolution. Identified trends can furthermore be used to estimate properties of planets that have not (and often cannot) be measured, which can help to select the most promising targets for time-consuming atmospheric characterization studies by instruments such as the James Webb Space Telescope. However, making full use of the archive has turned out to be challenging.

One of the principal difficulties is that the archive consists of thousands of planets, but each entry has recorded values for only a small subset of the measurable properties that varies depending on the discovery technique. For example, planets detected via the transit technique will have a measured radius, while the radial velocity technique provides a measured minimum mass. Similarly, direct imaging and gravitational microlensing techniques, which are sensitive to planets further from the host star, can measure a planet mass but include less orbital information due to being a single (or small section) snapshot of the planet trajectory. Host star properties are likewise sparsely measured, but their size and composition are expected to strongly impact the planet formation process [e.g. 103–107]. Detecting the same planet through multiple techniques can help fill these gaps, but is often not possible. For example, a transit observation requires the planet to pass across the star's surface as observed from Earth; an alignment that gets proportionally less likely for longer orbits (geometric probability decreasing as the inverse of the orbital radius, $p_T = R_\star/a$, for stellar radius $R_\star$ and average orbital distance, a). Likewise, stellar activity (such as star spots) is a continual bane for radial velocity detection, microlensing requires a one-off chance alignment with a background star, and imaging is currently most sensitive to very distant, young massive planets [108, 109]. The result is a large but sparse data archive of discovered planets, which is difficult to leverage to identify trends that simultaneously depend on multiple properties.

For this reason, attempts to construct relationships between planet properties are usually based on a small subsection of the discovered planets and involve just two properties, such as planet mass and radius, planet mass and stellar metallicity, or planet multiplicity and orbital eccentricity [110–114]. But inevitably, two-dimensional relationships cannot capture evolution pathways that depend on multiple factors, and nor can they utilize planets in the archive that lack either of the considered properties. The first issue can make it difficult to determine when a trend exists due to noise from other dependent parameters (see also section 4.3 and Figure 4.1). The latter point reduces the number of examples that can be included in the data analysis, and also risks restricting studies to planets with particular properties in common (such as planets on short orbits which transit to provide a radius measurements) and resulting conclusions may not hold more broadly through the exoplanet population.

Recently, Tasker et al. [115] (hereafter TLG2020) attempted to tackle the issue of multiple dependent planet properties by developing a neural network to impute missing values in the exoplanet archive. A strength of machine learning techniques such as neural networks is that multidimensional dependencies can be easily discovered in a dataset, allowing more complex trends to be leveraged when predicting planet properties. TLG2020 focused primarily on imputing missing planet mass and radius values, as the resulting average density is the most informative bulk property when considering planet composition or surface conditions [e.g. 116–118]. The accuracy of the TLG2020 neural network–a modified Boltzmann Machine (mBM)–was slightly better than two-dimensional imputations, and covered a wide range of masses. Additionally, the mBM produced a relative likelihood function for the planet mass or radius that could be sampled to produce a probability distribution. This was an informative way to explore the imputation, with peaks indicating when multiple planet sizes could be found at a particular orbit in similar planetary systems.

However, the mBM had a major limitation. The network had to be trained on a dataset where every entry had a complete set of properties, with no missing values. This significantly limited how much of the exoplanet archive could be used by the network to find the multidimensional connections between properties. In order to have a dataset large enough to be used for machine learning, TLG2020 restricted the included properties to planet mass, planet radius, orbital period, equilibrium temperature, stellar mass, and the number of known planets in the system. This resulted in a dataset of 550 planets with six properties. Other properties were sufficiently sparsely measured that their inclusion would have reduced the dataset size too significantly for meaningful results.

In this paper, the limitation of complete dataset training faced in TLG2020 is tackled by testing the ability of five different machine learning methods, all of which can leverage incomplete, multi-property datasets to impute missing information. This allows each algorithm to work with all currently known planets, rather than a small subset. The methods themselves are described in section 4.5, after first looking at visible two-dimensional trends in the exoplanet archive in section 4.3. The ability of these codes is compared with the TLG2020 mBM neural network by imputing planet mass and radius using the same complete dataset as in TLG2020 (section 4.6.1). This is then extended to all planets in the archive for the same six parameters (section 4.6.2), before finally including two more planet properties into the imputation (section 4.6.3). The results look at both the overall accuracy of the imputed planet properties, and what can be learned about the underlying demographics of the observed planet population based on the algorithm's imputed values. Finally, the most successful algorithm (the kNN×KDE) is used as a generative model to create a population of simulated planets which is analyzed to identify different planet groups whose properties are discussed (section 4.7). The overall findings, the added value of this approach in exploring the exoplanet archive, and how this can be used going forward to improve our understanding of planet formation are discussed in section 4.8.

4.3 The exoplanet data archive

Each of the algorithms in this paper utilizes a dataset of planet properties to impute missing values. Of course, this is only successful if the values in the dataset are related such that the known properties for a planet can provide information on the unknown values. As mentioned in the introduction, identifying complex relationships between multiple properties is challenging, but a sense of how pairs of planet properties are related can be gained by looking at the two-dimensional pairplots. This is shown in Figure 4.1 for eight planet properties. Diagonal panels show the univariate distributions as histograms, and off-diagonal panels show bivariate distributions as scatter plots.

The dataset for exoplanet properties used in this work is the NASA Exoplanet Archive[1]. Three different subsets of the full archive are explored in section 4.6 for imputing missing values. The first is the same dataset used in TLG2020 (pulled from the NASA Exoplanet Archive in 2018. Dataset: Tasker E.J. [119].), consisting of 550 planets each with six known properties: planet mass, planet radius, orbital period, planet equilibrium temperature, stellar mass and number of known planets in the system. The relatively small size of this dataset was due to the necessity in TLG2020 to use a dataset with complete properties and no missing values. The observed values for the planet properties in this complete dataset are shown in Figure 4.1 as black dots and histogram bars.

The next two datasets utilized in section 4.6 use all 5,243 confirmed planets in the NASA Exoplanet Archive (as of February 2, 2023). The majority of planets in these

[1] https://exoplanetarchive.ipac.caltech.edu/

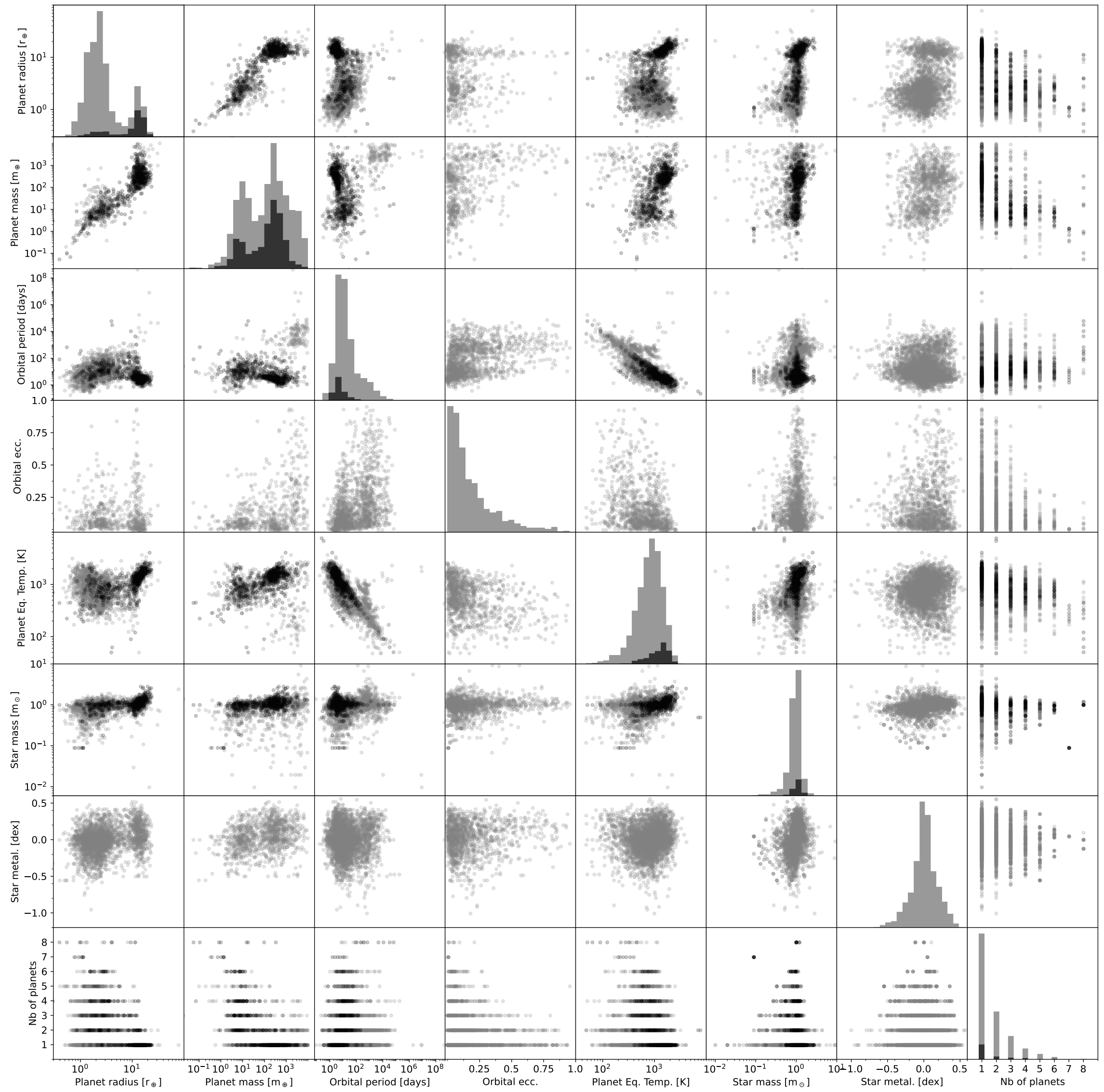


Figure 4.1: Pairplot for eight planet properties. Grey dots and histogram bars denote the full NASA Exoplanet Archive, while black dots represent the subset of planets used in the complete six properties dataset of TLG2020. Five variables (planet radius, planet mass, planet orbital period, planet equilibrium temperature, and stellar mass) have been log-transformed.

datasets have incomplete properties, which can be handled by the algorithms used in this work. The first of these datasets uses the same six planet properties as in the complete complete of TLG2020, while the second dataset additionally adds orbital eccentricity and stellar metallicity. The observed values in these datasets are shown as grey dots and histogram bars in Figure 4.1. All three datasets also include the eight planets of our Solar System (bringing the total number of planets in the second two datasets up to 5,251).

From the pairplot in Figure 4.1, a number of trends between pairs of variables are apparent. Unsurprisingly, planet radius is positively correlated with the planet mass, while the orbital period of a planet is negatively correlated with the planet equilibrium temperature. Notably, even these relationships show significant scatter, indicating other factors are playing a role. For example, planets on short orbits may have inflated atmospheres that result in a higher planet radius for a given planet mass, and stellar type obviously influences the equilibrium temperature on a given orbital period.

Looking at the planet mass and radius, it can also be seen that the subset of planets that was used in TLG2020 (marked in black) when a dataset of complete properties with no missing values was required, is not representative of the full population of confirmed exoplanets. In particular, comparison of the gray and black histogram for planet radius in Figure 4.1 (top left) shows that the majority of super Earths with radius $1\,r_\oplus < r < 5\,r_\oplus$ are missing from the complete properties dataset, despite being a dominant population in the archive overall. The more easily observed transiting hot Jupiters are therefore over represented in this dataset, which risks a strong bias in the resulting imputation.

Hints at less strong dependencies can also be seen in Figure 4.1. For example, planets in multi-planet systems tend to have smaller masses and radii. A similar trend was previously noted by Weiss et al. [120], who found that multi-planet systems discovered by Kepler often consisted of small ($r_p < 4\,\mathrm{R}_\oplus$), similarly sized planets (dubbed "peas in a pod"). Massive gas giants on short orbits are most likely to have migrated inwards, potentially disrupting other planets forming in the system by removing a substantial fraction of the planetesimal building material that would otherwise create rocky planets [121–123]. There is also likely an observational bias component here, as planets are more difficult to detect further from the star, so systems like our own with multiple cool gas giants, or those with smaller single planets orbiting further out, are not easily observable. Stellar mass shows a small trend with very large planet radii, although no notable pattern with smaller mass planets, which was also noted in regression models by Mousavi-Sadr et al. [124]. This may also be the effect of observational bias, since it is more challenging to find smaller planets around more massive stars, which will have a large ratio in their relative sizes.

The two additional parameters added to the third dataset in this study were orbital eccentricity and stellar metallicity. Evidence of trends between these properties and the original six planet properties can be seen, indicating that the addition has the potential to provide informative content. For example, eccentric orbits are more common for longer periods, as tidal interaction will act to reduce eccentricity at small separations from the host star. More interestingly, highly eccentric orbits are dominated by higher mass planets, and single planet systems. Planet multiplicity has previously been noted to correspond to low eccentricity, and it has been suggested that systems with a higher number of planets have suffered from less dynamical instability in the past, which would drive eccentric orbits [111, 125]. Meanwhile, massive planets may drive dynamic instability after the evaporation of the protoplanetary disc. In systems with multiple gas giants, this can lead to planet scattering that places one planet on an eccentric orbit and the other ejected out of the system (or on a distance orbit that is more difficult to detect) [126, 127]. This is possibly supported by a weak trend between stellar metallicity and orbital eccentricity, and between stellar metallicity and planet mass and radius. Metal rich stars are more likely to host massive planets, which in turn, may become dynamically unstable and create high eccentric orbits [128, 129].

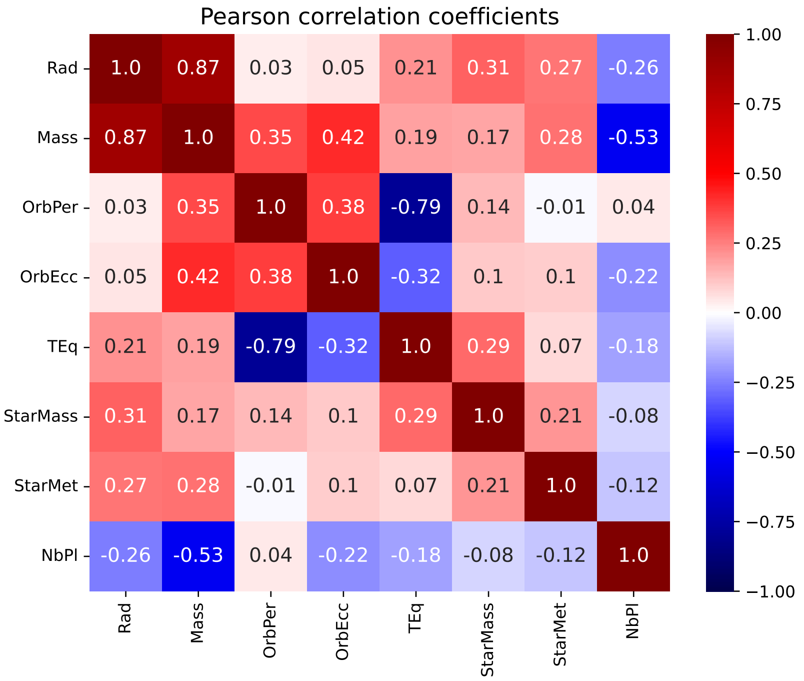


Figure 4.2: Pearson correlation coefficients for the eight chosen planet properties in the extended dataset. These values quantify the direction and the magnitude of pair-wise linear relationships between the exoplanet properties. Note that there exist no systematic way for interpreting Pearson correlation coefficients.

An attempt to quantify the trends between pairs of variables can be made by using the Pearson correlation coefficients, shown in Figure 4.2. The Pearson coefficient is a statistical measure of the linear correlation between two variables, taking a value between ± 1 [130]. The magnitude of the coefficient indicates the strength of the correlation, with a $+1$ or -1 value corresponding to a perfectly linear relationship, as can be seen along the diagonal of Figure 4.2 between identical properties. A coefficient value of 0 indicates that there is no linear dependency between the two variables. There is no objective rule to interpret Pearson correlation coefficients, but a general rule of thumb suggests that magnitudes between $0.2 - 0.5$ indicate moderate linear correlations, and magnitudes above 0.5 indicate strong linear corrections.

The two largest correlation coefficients in Figure 4.2 unsurprisingly correspond to the strongest visible trends in the pairplot in Figure 4.1. Planet radius and planet mass have a correlation coefficient of 0.87, indicating the very strong positive correlation between these two measurements of planet size. And the next largest is the negative correlation coefficient -0.79, marking the inverse relation between planet equilibrium temperature and orbital period. Weaker trends visible in the pairplot are also seen here at smaller coefficient values. Planetary systems with a high number of known planets correlate fairly strongly with lower planet masses, with a coefficient of -0.53. Stellar mass and planet radius have a weaker correlation of 0.31, reflecting that this trend is not apparent at all radii.

The correlation coefficients for orbital eccentricity and stellar metallicity support the evidence in the pairplot that the addition of these two properties should be important for predicting other values. A moderate coefficient value of 0.42 can be found between orbital eccentricity and planet mass, and slightly weaker values of 0.38, -0.32 and -0.22 between orbital period, equilibrium temperature, and number of planets in the system respectively. Stellar metallicity also has an indicated correlation with planet mass and radius (coefficient values of 0.28 and 0.27), supporting the trend that higher metallicity stars tend to host larger planets.

The multidimensional dependence of the planet properties is also clear from Figure 4.2. While a few combinations have very low correlation values with magnitudes less than 0.1, many pairs of properties show evidence of a moderate correlation. Similarly, very few pairs have extremely strong correlations, indicating that other processes are at play, and no single property is determining the value of another. This supports the idea that machine learning algorithms–with their ability to handle complex and highly dimensional dependencies–is a concept worth exploring in order to unlock the full potential of the exoplanet archive data.

It is worth noting that the trends visible in Figures 4.1 and 4.2 will include those due to observational bias. Both the transit and radial velocity techniques (the two most prolific planet hunting methods) favor closely orbiting planets, making it still challenging to survey longer periods. As a result, planet properties imputed from the archive should be considered "synthetic observations" that include observational bias. However, these are based on the only ground truth we currently have for planet formation: what we have observed.

4.4 Data cleaning

While as many observed values as possible for the considered six or eight properties were used in constructing the datasets utilized by the algorithms, values that risked being misleading were removed. In particular, values labelled as limits (suggesting the true value might be significantly larger or smaller) or stated without valid error measurements (indicating the value might not be an observed measurement) were not included in the final datasets.

As certain planets have been observed on multiple occasions, the NASA Exoplanet Archive offers several sets of observed parameters. Generally, the archive default dataset is used in this study. However, for missing values where another study has proposed a trustworthy (not a limit or without error bars) measurement, this value is included. As in TLG2020, missing planet equilibrium temperatures have been calculated using measurements of the host star radius R_*, the host star effective temperature T_*, and the average orbital distance a when available via $T_{\text{eq.}} = T_* \sqrt{R_*/(2a)}$.

Table 4.3 shows the chosen eight exoplanet properties considered in this study, and their missing rate, which is the fraction of planets that do not have a measured value. Since exoplanets have been discovered through various detection methods which provide different measured properties, the rate of missing values varies strongly across planet properties. Planet mass has a particularly high missing rate, since mass is not measured by the prolific transit technique (which accounts for roughly 75 % of planet detections),

Property	Units	Minimum	Maximum	Missing rate
Planet radius	$r_\oplus$	0.3	77.3	30.4 %
Planet mass	$m_\oplus$	0.02	9,852.0	72.8 %
Planet orbital period	days	0.1	402,000,000	3.7 %
Planet orbital eccentricity	.	0.00	0.95	70.1 %
Planet equilibrium temperature	K	48	7,719	13.5 %
Host star mass	$m_\odot$	0.01	10.94	0.5 %
Host star metallicity	dex	−1.00	0.56	10.1 %
Number of planets in the system	.	1	8	0.0 %

Figure 4.3: The missing rate (percentage of planets without a measured value) for exoplanet properties considered in this study in the NASA Exoplanet Archive. Most notably, 72.8 % of the known planets do not have a measured mass.

and only a minimum mass is measured by the radial velocity technique (approximately 20 % of detections). Measurements of the planet mass must therefore come either from a dual transit and radial velocity detection, or from the less common detection techniques, such as imaging or gravitational lensing. This is unfortunate, since as mentioned in section 4.2, mass is the most useful bulk quantity in accessing a planet's environment or interest in follow-up studies, yet it is difficult to measure during planet searches. For this reason, the present study focuses on the power of imputing planet mass as described in the next section. However, the algorithms used in sections 4.6.1 and 4.6.2 impute all properties that are missing.

4.5 Method and algorithms

The performance of five numerical data imputation methods are compared for imputing missing values in the exoplanet archive: the kNN-Imputer, MissForest, GAIN, MICE, and the newly proposed kNN×KDE. All five methods have the ability to utilize an incomplete dataset with missing values. In section 4.6.1, where a dataset of complete properties is used, performance is also compared with the modified Boltzmann Machine (mBM) neural network presented in TLG2020. In addition to imputing a single value for a missing entry, the kNN×KDE developed for this work capable of providing a probability distribution for the imputed value. Below, each of the numerical imputation algorithms are briefly described, along with hyperparameter values.

The kNN-Imputer uses the traditional k-nearest neighbors algorithm to fill-in missing observational values [76]. For each observation (planet) in the dataset with missing properties, the algorithm computes the distance to every other observation using the Euclidean distance between properties that are observed in both cases. The missing property is then imputed using the average of the k nearest neighbors that have that value observed. The latter step results in different neighbors being used to impute different properties, which can potentially lead to inconsistent values (e.g. an imputed planet mass and radius which lead to an unphysical density). Despite its simplicity, the kNN-Imputer has been shown to provide robust and accurate numerical imputation results [95]. The hyperparameter k which controls the number of neighbors can be optimized, and the value is fixed to $k = 15$ throughout this study.

MissForest is an iterative imputation algorithm which uses Random Forests for regression [78]. All missing values are initially filled with the column mean (planet property mean). MissForest then considers each property individually, and replaces the initial mean values with imputed values based on a Random Forest regression using all other properties. After performing this step for each property once, the process can be repeated again. Observed values always remain unchanged, while the missing estimates are updated. MissForest stops either after a fixed number of iterations have been performed or when the imputed values have sufficiently converged. This numerical imputation method can be computationally expensive, but has shown great practical results and is flexible to heterogeneous dataset types and structures [82]. The hyperparameter corresponding to the number of trees employed by the Random Forest algorithm is set to $N_{\text{trees}} = 20$.

GAIN (Generative Adversarial Imputation Nets) is an artificial neural network which revisits the GAN (Generative Adversarial Network) framework to impute missing values in numerical datasets [79]. Standard GAN models are composed of a generator and a discriminator. While the generator is tasked to output realistic observations, the discriminator aims at differentiating between real observations (from the dataset) and fake observations (synthesized by the generator). Unlike standard GAN models which generate entire observations, GAIN works on a cell by cell basis with the intent to fill-in missing values with credible synthetic data. This method claims state-of-the-art numerical data imputation results and has benefited from a lot of attention recently. However, recent benchmarks indicate that GAIN practical performances are mediocre when employed with real data sets [81]. GAIN additionally has many hyperparameters to tune, such as the batch size, the hint rate (fraction of correct labels to hint at the discriminator), the number of training iterations, and an additional weight parameter α used to help the generator to mimic original observed data. In this work, only the number of training iterations was optimized, which is the the most sensitive hyperparameter for GAN models. A value of $N_{\text{iter.}} = 2,500$ was selected for best performance.

MICE (Multiple Imputation Chained Equations) is another iterative imputation algorithm which uses linear regression [77]. Similar to MissForest, missing values are first filled with the column mean (i.e. the mean of the planet property), and the algorithm then loops over every column, one at a time. For a given column, MICE employs a linear regression to estimate missing values, unlike MissForest which uses Random Forests. After the linear regression, missing values estimates are updated while original values remained untouched. MICE stops either after a fixed number of iterations has been performed or when the missing values estimates have sufficiently converged. This algorithm is appreciated for its simplicity, its low computational cost, as well as its absence of any hyperparameters. However, it is not able to capture non-linear dependencies.

Finally, **the kNN×KDE** is a newly proposed imputation algorithm, inspired by the traditional kNN-Imputer, and specifically tailored to return probability distributions for each missing value in an incomplete dataset [101]. As with the kNN-Imputer, the kNN×KDE searches for neighbors to a planet with missing properties by considering the Euclidean distance between those properties that are observed. However, in contrast to the kNN-Imputer, the kNN×KDE looks only for neighbors which have observed values for all missing properties to be imputed, rather than considering the properties individually. This ensures that imputed values are consistent with one another. Probability distributions are modeled as a mixture of Gaussians via Kernel Density Estimates (KDE) where each neighbor value is additionally weighted by distance to give greater importance to planets whose known properties are in close agreement. The hyperparameter of the kNN×KDE is fixed to $\tau = 50^{-1}$ for the softmax temperature, which controls the tightness of the effective neighborhood around each observation. A higher value for τ would distribute weights more uniformly across all neighbors, while a lower value would distribute most of the weight to the nearest neighbor. In addition, a new hyperparameter is introduced for the kNN×KDE, which limits the total number of neighbors considered for imputation. Its value is fixed to $N_{\mathrm{cap}} = 20$ throughout our study. Capping the total number of neighbors prevents the algorithm from using distant and irrelevant observations, potentially leading to broad average in dense areas of the parameter space (e.g. hot Jupiters or super Earths). As the kNN×KDE returns probability distributions (and not point estimates), estimating missing values is done by computing the mean of the distribution when needed.

4.5.1 Including the minimum mass

After the transit technique, the most successful planet detection method is the radial velocity technique. A planet detected via the radial velocity method will have a measured minimum mass due to the unknown inclination of the planet orbit. The minimum mass is an important proxy for estimating the true mass, but as it is a lower limit on the actual value, it cannot be passed to the algorithms as one of the measured planet properties. For the algorithms that return a single value, a minimum mass observation can therefore not be included when imputing missing values. However, the additional information from the minimum mass can be leveraged with the kNN×KDE by computing the convolution of the estimated probability distributions with the distribution of possible masses based on the minimum mass observation.

In section 4.6, the performance of the algorithms is tested on synthetic radial velocity observations, where both the measured planet radius and mass are concealed and a minimum mass measurement is generated. Following the same methodology as TLG2020, 100 minimum masses are generated for each planet with a missing mass and radius value by randomly drawing orbital inclination values following a sinusoidal distribution. The convolution between the estimated mass distribution from the kNN×KDE algorithm and the distribution of possible masses given an observed minimum mass is performed as described in Tasker et al. [115]. In practice, this corresponds to computing and assigning new weights to the individual mass-radius bivariate samples returned by the kNN×KDE.

The final mass and radius estimates by the kNN×KDE are computed by taking the average of the distribution after convolution. Because this procedure is repeated for 100 different possible minimum mass measurements, each planet has 100 pairs of mass and radius estimates. The final estimate is therefore taken as the mean over the 100 estimates. Note that the quoted error for mBM in section 4.6.1 differs from that in TLG2020 due to a small difference in the definition of the overall error that better matches with the plotted data. Instead of taking the average over the 150 planets and the 100 convolutions in a single step, the average over the 100 convolution is performed first to provide the mass and radius estimates. The RMSE is then computed over the 150 planets in the test set. This way of computing the error more faithfully reflects the data, particularly the visual data plotted in Figure 4.6 which already represent the average taken after 100 convolutions.

Of course, a real observation would only include a single minimum mass measurement. The resulting error might therefore be significantly greater than our average, depending on the degree of inclination of the planet orbit.

4.6 Imputing planet properties

The five machine learning algorithms presented in Section 4.5 draw on different techniques to impute missing values in a dataset of items with multiple potentially related properties. Unlike the modified Boltzmann Machine (mBM) neural network presented in TLG2020, each of these five techniques has the ability to utilize an incomplete dataset of properties for the imputation of the missing values. This removes the substantial restriction of only presenting the algorithm with a small complete subset of the available data from which to calculate missing values.

As described in section 4.3, the performance of the algorithms are tested on three datasets. The first is the same dataset that was used in TLG2020. This is a subset of 550 planets with observed values for six properties: planet radius, planet mass, orbital period, planet equilibrium temperature, stellar mass and the number of known planets in the system. The small size of this dataset–only about a tenth of the known planets today–is due to the requirement of the mBM in TLG2020 for a training dataset with no missing properties. The same dataset is initially used to compare performance between all algorithms, and to assess the difference in using complete versus incomplete data.

The second dataset includes the same six properties but no longer requires every planet in the dataset to have a complete set of observed values. This greatly expands the dataset size to 5,251 planets (the full set of discovered planets listed by the NASA Exoplanet Archive on February 2, 2023, including Solar System planets). The third dataset contains the same planets but now adds two additional properties to be used in the imputation: stellar metallicity and planet orbital eccentricity.

For each dataset, the algorithms are tested by artificially concealing known values and imputing these with each code. To replicate two of the most likely use cases, planet mass was redacted followed by a second test in which both planet mass and radius were removed from a test set of planets. The first case replicates an observation performed with the transit technique, which is the planet detection method employed by the dedicated space-based planet hunting missions and accounts for over around 75% of the planet discoveries. The transit technique measures a planet radius, but not planet mass. The second test resembles an observation using the radial velocity method, which provides a minimum mass for the observed planet, $m_m = m_p \sin(i)$ for orbital inclination, i, and no radius measurement. Both planet mass and planet radius are therefore initially concealed and imputed by the algorithm. The minimum mass observation is then leveraged by convolving the probability distribution of possible mass and radii values with the distribution of possible masses from the minimum mass measurement (see section 4.5.1). This was only possible for the kNN×KDE algorithm and the previous mBM code, which can produce distributions (rather than single value estimates) for imputed properties.

4.6.1 The complete properties dataset

The performance of the five algorithms was first tested on the 550 planet dataset used in TLG2020. Each planet entry in that dataset had six measured properties for planet radius, planet mass, orbital period, planet equilibrium temperature, stellar mass and the number of known planets in the system, with no missing values. 150 planets within that dataset were assigned as a test set, with selected properties artificially hidden and imputed by each algorithm. To compare directly with the mBM performance in TLG2020, the same 150 planet test subset that was presented in the main results section in TLG2020 is used to test the five proposed algorithms.

As the new algorithms can leverage incomplete data to estimate missing values, all 550 planets in this dataset were provided to each imputation method (where 150 planets had one or two missing property values) to estimate the missing properties. This differs from TLG2020, where the mBM network needed to be trained on the 400 planets with complete properties, and the resulting relative probability density functions created by the mBM was then used to impute the missing properties in the test set.

Mass prediction in the transit regime: complete properties dataset

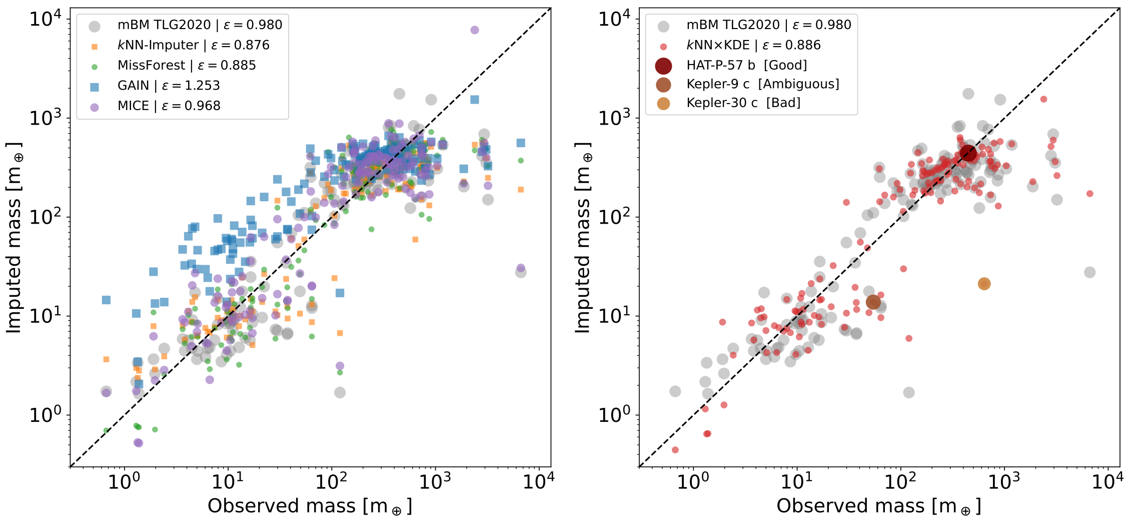


Figure 4.4: Test results when using the complete properties dataset where the 150 test planets are treated as transit observations, with missing mass values. The left-hand plot shows four proposed imputation algorithms alongside the mBM code in TLG2020. The right-hand plot shows the comparison between the observed mass and imputed mass for the mBM code and the kNN×KDE algorithm. The figure legend shows the average error across all 150 plotted planets. The diagonal dashed line marks a perfect correspondence between the observed and imputed values. The distributions of the three planets marked in the right-hand legend are shown below.

Figure 4.4 shows the imputed mass compared to the observed mass value for the 150 test planets when their observed mass value was concealed and imputed by each algorithm.

This simulates the performance for estimating mass from a transit detection, which can provide five of the six dataset properties, but no mass measurement. The left-hand plot of Figure 4.4 shows the performance of four of the new codes compared with the mBM of TLG2020 shown in grey. On the right-hand plot, the kNN×KDE algorithm marked in red is compared with the mBM.

For each planet, p, that has an imputed mass, the error is computed by taking the natural logarithm of the ratio of the observed planet mass ($m_{p,o}$) and the imputed mass ($m_{p,i}$) to give $\epsilon_{\mathrm{p}} = \ln\left(m_{p,o}/m_{p,i}\right)$. These values are averaged over the 150 test planets, N, by taking the root mean square to give the final reported error for each algorithm as $\epsilon = \sqrt{\left(\sum_{p=1}^{N} \epsilon_p^2\right)/N}$. This error value is shown in the top left corner of Figure 4.4 for each of the five algorithms and the mBM. A perfect match, with an error of $\epsilon_p = 0.0$, would lie along the diagonal dashed line.

A comparison of the average error from all five codes and the mBM shows that four out of the five new imputation techniques surpass the original result, suggesting that the ability to train on the properties available for all 550 planets (excluding the 150 planetary mass values) was a benefit to the imputation. Two main groups of planets can be seen in the distribution of planet masses, one in the gas giant regime (with masses similar to that of Jupiter) and the second in the smaller super-Earth regime, with masses about ten times that of the Earth. Planets do exist between and outside these two groups, but are less clustered. All algorithms perform best in the mass regime of Jovian planets. This is not surprising, as large gas giants are typically easier to detect than small rocky worlds, providing a more densely packed parameter space over that mass range.

The poorest performance was that by the GAIN algorithm, which can visually be seen as an average overestimate of the planet mass over the full range of masses in the dataset. In spite of impressive results provided by deep-learning models for image, video, or text data, it has been shown that statistical methods (and in particular tree-based models) remain the state-of-the-art for numerical and tabular datasets [82]. This state of affairs is observed here, where the kNN-Imputer and the kNN×KDE (nearest-neighbor methods), as well as MissForest (a tree-based model), providing with best results, with an error of around $\epsilon = 0.88$ that corresponds to a factor of 2.4 from the observed mass. Statistical tools have very few hyperparameters to train which not only prevents from overfitting (i.e. when the model predictions are accurate only for the training dataset), but also facilitates results interpretation. Conversely, the GAIN is a generative adversarial network which necessitates to fine-tune thousands of trainable parameters. Generative adversarial networks are known for being particularly hard to train, to interpret, and to diagnose [131]. Notably, they easily suffer from "mode-collapse problem", where the output distribution shrinks down to a small region of the desired target distribution. This is what is happening here: the dominating Jupiter planets mislead the GAIN into generating planets mostly in this regime, therefore over estimating the mass of planets in the Super-Earth regime.

In addition to a single imputed mass value, the kNN×KDE algorithm can provide a probability distribution for the imputed value thanks to weighted kernel density estimates. This can be used to understand the origin of the estimated value, which can reveal information about the underlying demographics of the observed planet population that is harder to decipher from two dimensional trends alone. To assess the value of this addi-

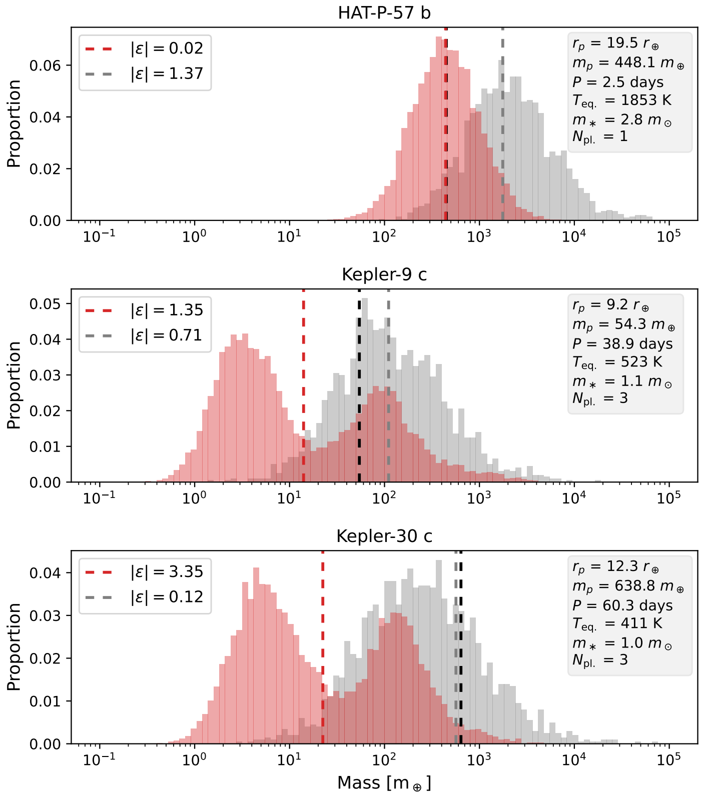


Figure 4.5: Distributions of the imputed mass values calculated with the kNN×KDE for the three planets highlighted in Figure 4.4. The red histogram shows the distribution calculated by the kNN×KDE, and the gray histogram is the distribution from the mBM in TLG2020. The vertical black line is the observed mass for the planet, while the red and gray vertical lines show the imputed value from the kNN×KDE and the mBM, respectively. Top panel shows the mass distribution for HAT-P-57b: a hot Jupiter with a low error for the imputed mass. The middle and lower panes show the mass distributions for Kepler-30c and Kepler-9c, both of which have higher errors on the imputed value.

tional feature, the mass distributions for three planets imputed by the kNN×KDE (red histogram) and the previous distributions from the mBM in TLG2020 (gray histogram) are shown in Figure 4.5. These are the same three planets indicated by enlarged circles on the right panel of Figure 4.4 and were selected to better understand their error value.

The top-most pane in Figure 4.5 shows the mass distribution for HAT-P-57b, which has one of the lowest errors in the dataset. HAT-P-57b is a hot Jupiter, with a mass of $1.41\,\mathrm{M_J}$, radius of $1.74\,\mathrm{R_J}$ and orbital period of just 2.5 days [132, 133]. Hot Jupiters were among the first extrasolar planets to be discovered, as both the radial velocity and transit techniques are most sensitive to planets with large sizes and short periods. Because of this observational bias, hot Jupiters are well represented in the exoplanet archive, despite only orbiting about 1 % of stars [134]. This is particularly true of this complete dataset, as the requirement to have both planetary mass and radius measurement usually requires both a transit and radial velocity detection, bumping the occurrence rate of the more easily detected Jovian-sized planets ($\mathrm{M}_p > 0.1\,\mathrm{M_J}$) within the database to 72 %, 86 % of which have orbits shorter than 10 days.

As a result, HAT-P-57b sits in a dense area of the parameter space for all six of the planet's observed properties (this can be seen visually by estimating the planet's position on Figure 4.1) and the imputed values for hot Jupiters typically have the lowest errors across all the algorithms. The relatively narrow profile width indicates that the 20 nearest neighbors found by the kNN×KDE all have similar masses. This suggests HAT-P-57b belongs to a typical class of known exoplanets, and the algorithm is confident in favoring a Jovian mass world based on the five provided measurements. The previous estimate by the mBM for TLG2020 also predicted that HAT-P-57b would be Jovian size, but estimated a larger mass with a broader distribution of possible values.

The middle and lower panes of Figure 4.4 show the mass distribution for planets with higher errors, with Kepler-30c (lower pane) having a particularly large error. In both cases, the reason for the high error is that the imputed value is an average between two possibilities for the mass. Based on the five observed properties available to the algorithm, the kNN×KDE therefore considers that Kepler-9c and Kepler-30c could be either gas giants with a mass around $0.5\,\mathrm{M_J}$, or super Earths with a mass around $3\,\mathrm{M_\oplus}$.

For the kNN×KDE, this dichotomy highlights one of two situations. Either the five measured properties of Kepler-9c and Kepler-30c can belong to two different mass regimes of planets, or the pair are unusual within the exoplanet demographic and their twenty neighbors are covering a wide area of the parameter space. In this case, a comparison with the planets' observed properties listed in Figure 4.5 with the demographics pairplot in Figure 4.1 reveals that it is the second option: both planets are a little unusual.

The size of Kepler-9c is rare within current exoplanet discoveries, consistent with a inflated sub-Saturn [135]. This places Kepler-9c between the two most common sizes for discovered extrasolar planets, sitting at the dip in the radius histogram in Figure 4.1, and the thin neck of the planet radius versus planet mass plot. This parameter region is particularly sparse for the complete dataset used here (black dots in Figure 4.1) where very few examples of planets exist with radii between the super Earth and gas giant regime. It is therefore not surprising that the kNN×KDE algorithm has found that the closest matches the planet's known properties have masses both higher and lower than the ground truth.

Despite being aware of more larger radii planets than small, the kNN×KDE favors the smaller super Earth mass peak for Kepler-9c. This is probably because Kepler-9c resides in a system with three known planets. Multi-planet systems have been found to favor similar sized worlds with smaller sizes [120], reducing the probability that such planets are gas giants. The nearest neighbors to Kepler-9c are therefore more likely to have lower masses.

One of the main advantages of a returned distribution is the ability not to settle for the returned imputed value. In the case of a multi-modal profile, it does not make sense to select the average value as the imputed mass, but rather select one of the peaks. Given the large radius for Kepler-9c and the reasonably similar size of both mass peaks, a researcher wishing to estimate the mass value might select the higher mass peak as the most probable value, but note that the presence of the second peak meant the planet's size in a multi-planet system was rare. In this case, the resulting estimation would be close to the observed value.

The origin of the bimodal profile for Kepler-30c is more intriguing. Unlike Kepler-9c, the planet's observed radius is not ambiguous: at $1\,\mathrm{R_J}$, Kepler-30c is a gas giant, although its observed mass at $2\,\mathrm{M_J}$ is at the upper end for planets of that size [136]. Given the certainty for HAT-P-57 b, it therefore seems initially surprising that the resulting mass profile do not strongly peaks at a gas giant mass. However, the orbital period for the planet at 60 days is quite long and past the typical distance for hot Jupiters, which usually have orbital periods less than about 10 days. As can be seen from the Figure 4.1, this 60 day orbit makes the planet a more unusual find, and the surrounding parameter space in that property is relatively sparse, with the majority of planets at that distance having super Earth masses.

Despite the unusual period, there are sufficient planets close to Kepler-30c in the visible five properties for the kNN×KDE algorithm to create a relatively close group in that parameter space with which to estimate the mass. But the mass for those neighbors turns out to cover a large range than for the other properties, resulting in the bimodal profile with the peak at super Earth masses. This indicates that planets on these longer more temperate orbits have a surprisingly wide range in density. It is an interesting trend to note, and potentially might suggest that this group of planets have a range of compositions driven by their evolution, which might include migration from the outer, icy parts of the protoplanetary disk, or in-situ formation. However, this might also be due to the challenges of detecting planets at these distances from the host star. Many planets in multi-planet systems on longer orbital periods like Kepler-30c have their mass measured via transit timing variations (TTV). TTV can often gives quite high errors on the mass, which may explain the wide range in densities amongst Kepler-30c's neighbors, rather than strong variations in their mineralogy.

The true observed mass for Kepler-30c is at the highest end of the imputed distribution. There is a definite bump around that location, but it is not considered the most probable result. In this case, manually selecting a peak is not enough to get an accurate imputation, but the presence of multiple peaks is still a flag to investigate the origin of the imputation which reveals the uncertain demographics of the surrounding population.

Interestingly, the previous mBM model is broader and not bimodal for either Kepler-9c or Kepler-30c, considering both planets likely to be gas giants. This may be because the neural network has internally weighted the importance of radius measurement in indicating mass more highly than other properties. In these cases, that produces a more accurate answer, but the neural network is more opaque than the statistical kNN×KDE, so reveals less about the underlying planet demographics.

Mass and radius prediction in the RV regime: complete properties dataset

The second test using the six complete properties dataset simultaneously conceals and imputes both the mass and radius values, and then weights the resulting distributions with a minimum mass measurement as described in section 4.5.1. This replicates imputing mass and radius values for a radial velocity observation, where a minimum mass, orbital period, effective temperature, stellar mass, and number of known planets would be expected data, but no radius or true mass measurement. Since the convolution step with the minimum mass is only possible where a distribution of values can be imputed, only the kNN×KDE algorithm can be compared with the previous mBM neural network for this test.

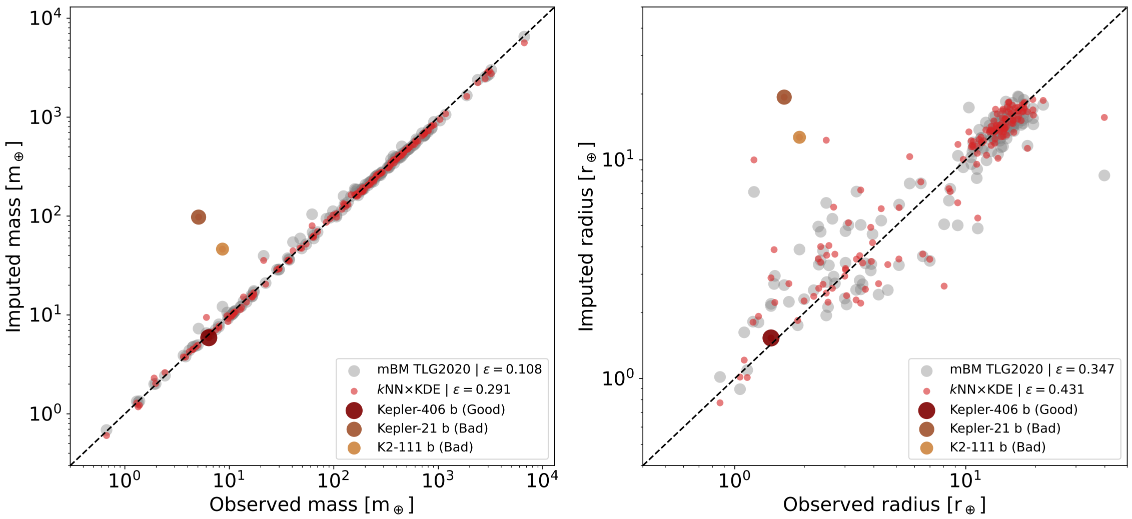


Figure 4.6: Test results when using the complete data set where the 150 test planets are treated as radial velocity observations, with missing radii values and a minimum mass measurement. The left-hand plot shows the results for the mass imputation, after the imputed distribution has been weighted by the minimum mass. Red dots show the kNN×KDE algorithm results, while the light grey distribution is from the mBM code in TLG2020. The right-hand plot shows the comparison between the observed radius and imputed radius values that correspond to the weighted imputed masses. The mass and radii imputed value distributions for the three highlighted planets are shown in the next figure.

The left-hand plot in Figure 4.6 shows the imputed mass versus the observed mass for the same 150 test set of planets as in section 4.6.1, where this time both the mass and radius values have initially been concealed before weighting with a minimum mass value. The low error and tightness of the fit compared to the transit regime test in Figure 4.4 is a reflection of the importance of the minimum mass measurement compared to radius in imputing the planet mass. We can compare this to the right-hand plot of Figure 4.6, which shows the radius imputation where we have a minimum mass measurement. The scatter here is visually similar to Figure 4.4, where the mass is being imputed with radius information.

The average error, ϵ, for the kNN×KDE is higher than the mBM in TLG2020. The error is exacerbated by particularly poor imputations for the mass and radii of Kepler-21b and K2-111b, which are highlighted in larger circles in Figure 4.6, and whose mass and radii distributions imputed by the codes (prior to weighting by the minimum mass distribution) are shown in the second and third panel of Figure 4.7. From the distributions, it is evident that both kNN×KDE and the mBM believe that these two planets should have a higher mass and radius than observed. In fact, the two codes strongly agree with one another that the expected masses and radii are those of a Jovian-sized gas giant. This

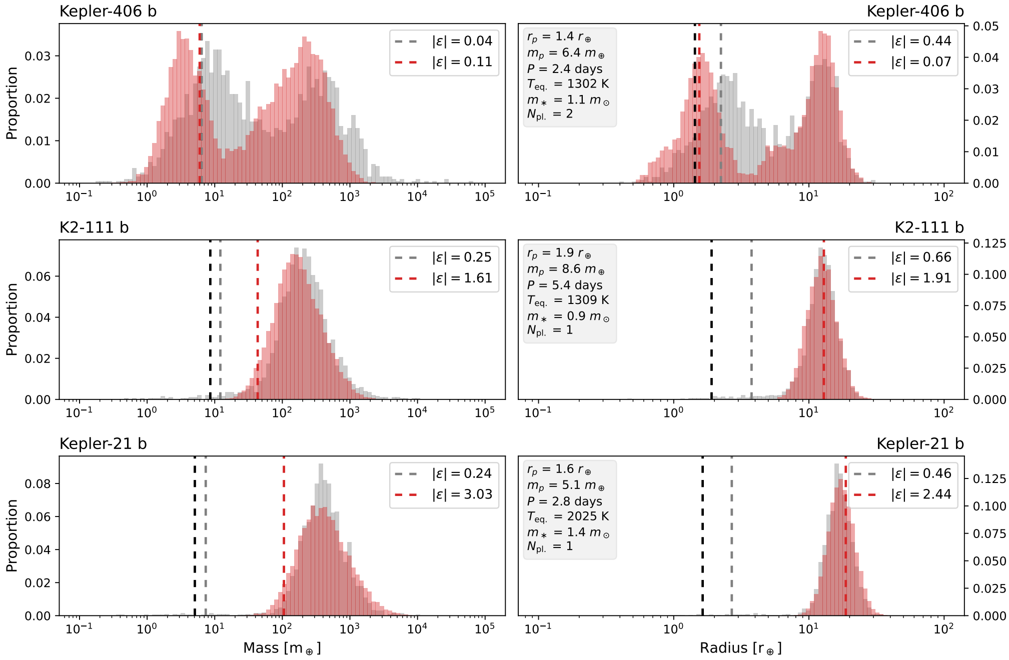


Figure 4.7: Distributions of the imputed mass and radius values calculated with the kNN×KDE for the three planets highlighted in Figure 4.6. The red histogram shows the distribution calculated by kNN×KDE, while the gray histogram is the distribution from the mBM in TLG2020. The distributions are from the initial imputation of the algorithm, before the weighting with the minimum mass distribution. The vertical black line is the observed mass for the planet, while the red and gray vertical lines show the imputed value from the kNN×KDE and mBM, respectively. The top panels show the distribution for Kepler-406b, which has a very low error. The next two distributions fro K2-111b and Kepler-21b both have high errors.

indicates that there is something unusual about finding a planet of smaller size in these particular environments.

The unusual nature of K2-111b can be understood by comparing the distribution to that of a very low error planet, Kepler-406b (top pane in Figure 4.7). Unlike the distributions for K2-111b or Kepler-21b, both kNN×KDE and mBM propose two almost equally possible options for the mass and radius of Kepler-406b: one Jovian-sized planet and one super-Earth possibility. With neither a planet mass nor radius measurement available to the codes, the imputation is based on the planet's orbital period, equilibrium temperature, stellar mass and number of known planets in that system. In three of these properties, both the low error Kepler-406b and the high error K2-111b are very similar. However, Kepler-406b was known to be in a two-planet system, whereas K2-111b was thought to be alone. As was discussed in section 4.3, the bottom row on the pairplot in Figure 4.1 shows a trend towards massive and closely orbiting planets in single planet systems where a migrating hot Jupiter may have suppressed additional planet formation, but a more even range of masses where two planets are present. The appearance of the second, lower-mass peak for Kepler-406b was therefore driven by the inclusion of the second system planet. In this situation, the minimum mass measurement resolves the dichotomy, causing the algorithm to select the lower mass peak as the most likely value.

These two distributions for Kepler-406b and K2-111b indicate that both super Earths and gas giants are commonly found on orbits of a few days around solar-type stars. However, in the case where only one planet orbits, a hot Jupiter whose migration is capable of disrupting further planet formation is more common. This then leaves the question as to why K2-111b is not a gas giant, but has a measured mass of a super Earth. The answer is that the data for the K2-111 system is actually incomplete. The dataset used in this section was the same dataset for TLG2020, and taken from the exoplanet archive in 2018. In 2020, a second planet was discovered orbiting K2-111 [137]. The presence of this second planet was therefore hinted at by the high error of the kNN×KDE and mBM result based on the demographics of the planets in the dataset, as the mass of K2-111b would be more commonly found in a multi-planet system. This demonstrates that numerical imputation schemes that can return distributions such as the kNN×KDE can be used not only to impute values, but also to investigate known planet properties to see how typical they are amongst the known exoplanet population. It is information that could be useful for plans for follow-up discovery missions such as the proposed *Japan Astrometry Satellite Mission for INfrared Exploration* (JASMINE), which will search for undiscovered planets in known systems [138].

Both the kNN×KDE and the mBM also mistook Kepler-21b for a gas giant, rather than the observed super Earth. It is possible that this system also has a second, undiscovered planet. But in this case, the planet also has other properties that also make its small size unusual. Kepler-21b closely orbits a very bright F-type star, giving the planet's properties a particularly high stellar mass and equilibrium temperature within the current demographics [139]. Looking at the pairplot in Figure 4.1, a trend exists between stellar mass and planets with large radii. There is also a weaker hint of a trend between planet mass and equilibrium temperature, with hotter planets typically being more massive. This results in the planets with closest matches to stellar mass and equilibrium temperature (neighbors in the parameter space) to Kepler-21b being gas giants. As mentioned in section 4.3, this trend may be due to the difficulty in finding smaller planets around more massive stars. The situation is particularly marked for the complete properties dataset used in this section, which requires a transit observation for every planet to measure the planet radius; a technique with a strong bias towards smaller ratios for the star to planet radius. It therefore seems likely that even a second planet in the system would not be sufficient to make Kepler-21b “normal" within the exoplanet population and topple the likelihood that Kepler-21b is a gas giant. Instead, the high error reflects an unusual outlier in the exoplanet demographics, either due to observational constraints or a lack of examples for planets around F-type stars.

With the two profiles strongly overlapping, it is initially surprising that the mBM network does not also have a high error for these two planets. However, the mBM shows heavy-tailed distributions with a few outlying values at the lower end of the distribution. This tail artificially allows the convolution with the minimum mass measurement to favor an estimate close to the actual value, even when it lies in a very low probability region of the imputed distribution. In the case of the kNN×KDE distributions, the twenty neighbors all have high masses and radii, and a tail is not produced. The minimum mass therefore falls completely outside the range of estimated possible values. In this situation, the convolution with the minimum mass cannot sufficiently influence the final imputed value. The distance between the minimum mass value and the distribution peak can also indicate the high error for planets such as K2-111b and Kepler-21b even when the distribution shape looks reasonably certain. The minimum mass not lying within a high probability area of the distribution flags a disagreement between the two mass distributions, unless the orbit is especially inclined. This allows a similar investigation as above, even in cases where the true mass is not known. Conversely, a situation like Kepler-406b, where the minimum mass lies close to a mode of the mass distribution, would be expected to increase the likelihood of a low error.

For this complete properties dataset, four out of the five new algorithms performed comparably on average with the mBM neural network, typically producing a slightly lower error result. This indicates that the ability to train on incomplete data does not degrade the performance where a complete set of properties is available. For the kNN×KDE where planet property distributions could be generated, the results frequently resembled the mBM, indicating that these two independent methods were extracting similar conclusions about the exoplanet demographics. In areas where the two differed, the kNN×KDE and mBM alternately achieved the better result. The origin of the kNN×KDE numerical algorithm is easier to understand in comparison with the pairplot in Figure 4.1, but the mBM neural network can preferentially weight properties it deems more valuable. These are considerations when deciding what kind of code to use.

In both the kNN×KDE and mBM case, the distributions of the imputed properties provide significantly more information than the imputed value alone. The distribution shape can indicate whether a broad range of values are consistent with the observed planet demographics, or if distinct choices (multi-modal distributions) are more likely. The distribution can also identify outliers, and help to access whether a planet's unusual properties might be due to a rare discovery or an incorrect value (such as a missing planet). In the case of radial velocity observations, examining the distribution also distinguishes between an imputed planet size driven by the minimum mass observation, versus one where the imputed distribution from the code and that of the minimum mass have strong agreement.

4.6.2 The full archive dataset: six properties

The dataset used with the five algorithms is now extended from the 550 planet subset with six complete properties to using an incomplete dataset with missing values that includes all 5,251 planet discoveries. The factor ten increase in planet number also greatly increases the range in properties. While the complete properties dataset covered a mass range of about $1\,\mathrm{M}_\oplus$ to $5{,}000\,\mathrm{M}_\oplus$, the full archive runs from approximately $0.1\,\mathrm{M}_\oplus$ to $10{,}000\,\mathrm{M}_\oplus$ (about $31\,\mathrm{M}_\mathrm{J}$: into brown dwarf territory). The mBM presented in TLG2020 is not able to train on incomplete data, so it is dropped from the comparison at this point. Without the necessity of a separate training set that was required by the mBM, the performance of the five algorithms is tested using a leave-one-out cross-validation method. This consists of removing one or more properties from one planet entry and imputing those missing values based on the remaining observations in the dataset. The process is repeated for each planet in the dataset. This method makes full use of the observed data, and prevents relying on a particular train/test data split.

As in the previous section, algorithm performance is tested in the “transit regime" where known observed mass value are redacted and imputed to replicate the use case for a transit observation, and the “radial velocity regime", when both mass and radius values are imputed with a minimum mass guide.

Mass prediction in the transit regime: full archive dataset

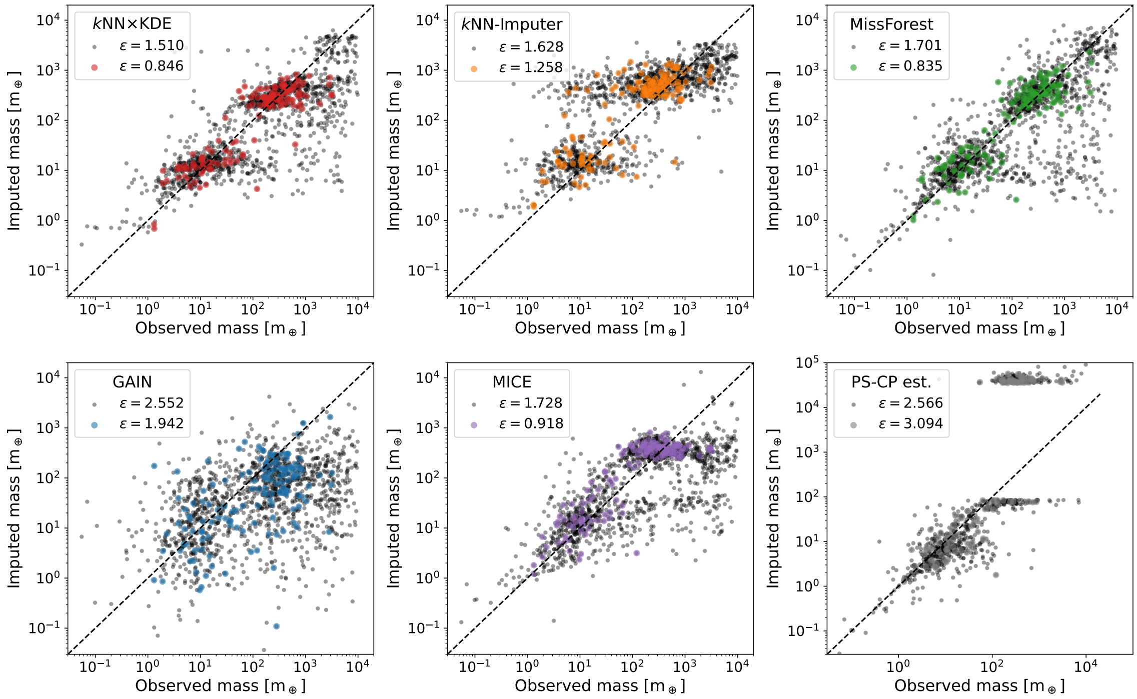


Figure 4.8: Test results when using the full exoplanet archive in the “transit regime" where planet mass is concealed and imputed. Each of the five algorithms imputes the planet mass for planets with observed mass value, treating each planet as if it had been detected as a transit observation, with a radius observation recorded where available. The final pane shows the results of the mass–radius relationship of Chen and Kipping [110] used by the Planetary Systems Composite Parameters (PS-CP) table of the NASA Exoplanet Archive. The black dots in each panel show the entire database. The colored dots correspond to the planets in the same test set presented in section 4.6.1 for comparative purposes. The two error values are for the complete dataset and the previous test subset.

Figure 4.8 compares the performance of the five algorithms in imputing planet mass. For each planet with a recorded mass observation (a total of 1,426 planets in the full archive dataset), the planet mass is concealed from the algorithm which then imputes the value based on the remaining observed properties for that planet and the proprieties of the other planets in the dataset. Note that while this test is referred to as a synthetic transit observation, the incomplete dataset means that it is no longer true that the imputed planet will always have a radius measurement.

Due to the number of planets now being tested, Figure 4.8 shows the results from each algorithm in a separate plot. Black dots show the imputed mass values for all planets tested, while the colored dots represent the planets that were also in the test set for the complete dataset in section 4.6.1. As it is no longer possible to compare with the mBM, the last panel in Figure 4.8 shows the results of the mass-radius relationship of Chen and Kipping [110]. The mass-radius relationship uses a piece-wise linear function which links mass and radius over scales from small rocky planets to stars. It is a rapid scheme that is included by default in the NASA Exoplanet Archive for the Planetary Systems with Composite Parameters (PS-CP) Table to provide an estimate of one of missing mass or radius.

The scatter in Figure 4.8 is generally higher than that for the complete dataset in Figure 4.4 and reflected in a higher average error, ϵ. This is unsurprising, as many planets now have less observed properties from which to estimate the mass which is resulting in higher errors. The algorithm with the lowest overall error is the kNN×KDE, with the GAIN once again performing most poorly. With the inclusion of the full archive to now assist the imputation, the error for the original planets in the test set of TLG2020 (denoted by colored dots) has also changed. For kNN×KDE, MissForest and MICE, the addition of the full dataset has improved the mass imputation for the original test planets, but the error has actually increased significantly for kNN-Imputer and GAIN. In the case of kNN-Imputer, the inclusion of the full dataset is resulting in a relatively strong bias towards predicting an average mass for the high mass planets. This is visible as a horizontal plateau at around a Jupiter mass. The same plateau is also visible in the MICE algorithm and, to a lesser extent, in the kNN×KDE plot. The kNN-Imputer algorithm also shows evidence for a second plateau for lower mass planets, effectively selecting one of two average masses for the majority of the planet population, which explains the high average error.

The emergence of the averaged plateaus depends on how each algorithm is imputing the single point missing values. Shown in the upper-left of Figure 4.8, the kNN×KDE calculates imputed values by taking the mean of the probability distribution. This mean is affected by the number of neighbors the algorithm draws upon for the distribution. As described in section 4.5, the number of neighbors is determined by the newly added hyperparameter N_{cap}. The value of N_{cap} needs to be large enough to accurately sample the parameter space in the vicinity of the planet whose properties are being imputed and provide an informative probability distribution. But for a single point imputation, N_{cap} should also be sufficiently low that all included neighbors have observed properties close to those of the imputed planet. If the distribution draws from too many neighbors, the mean becomes influenced by potential outliers near the distribution tails, leading to broad averages. Conversely, a lower neighbor number visibly reduces any plateau formation, but at the cost of a higher scatter and higher average error ϵ: this is the famous bias-variance trade-off. Here, a value of $N_{\mathrm{cap}} = 20$, was selected as a compromise between error performance on the average imputed value, and an informative distribution. Note that an average imputed value is computed primarily for the purpose of code comparison. However, as seen in section 4.6.1, a more thorough and interesting exploration of the exoplanet comes from studying the origin of the probability distributions for the imputed value. If one is not interested in a single imputed value, then plateau formation is not a problem since it originates from averaging over the distribution of choices for the mass.

Surprisingly, the kNN-Imputer (upper middle panel of Figure 4.8) shows a much stronger plateau than the kNN×KDE even though the scheme uses less neighbors ($k = 15$, see section 4.5). This is because the kNN-Imputer imputes the missing values for each planet individually, one after another, unlike the kNN×KDE that imputes all the missing values for each planet together. As selected neighbors are required to have observed values only for the missing property being currently imputed, imputing one property at a time allows the kNN-Imputer to select from a much larger pool of surrounding planets. This potentially unlocks access to more relevant neighbors, but also increases the risk for biased imputation, where the same neighbors are repeatedly used in denser areas of the parameter space. This is what is happening with the formation of the two mass plateaus at Jupiter-sized planets and at around $10\,\mathrm{M}_\oplus$. Increasing the neighbor number for the kNN-Imputer would lead to even stronger plateaus. Visually removing the averaged plateaus requires dropping below $k = 10$ neighbors for imputation, but the scatter is much higher with such a low number of neighbors, and the overall error also higher (lower bias implies higher variance in the prediction).

The apparent plateau for the imputed planet masses by the MICE algorithm (lower middle panel of Figure 4.8) is also a consequence of a dense gas giant parameter space region. However, this problem cannot be addressed here as MICE has no hyperparameter to modify its behavior. Because MICE uses linear regressions to estimate missing values, the dense collection of observed masses in the Jupiter-sized regime draws mass estimates towards the broader average. In particular, MICE struggles to predict planet masses above $1{,}000\,\mathrm{M}_\oplus$. That said, it is worth noting that the average plateau effect does not happen in the super-Earth regime, probably because the observed masses span a broader range, and also because other observed properties are more diverse in the super-Earth regime than with hot Jupiters.

Interestingly, MissForest do not show any plateau (upper right panel of Figure 4.8). As described in section 4.5, Random Forests–that are used by MissForest for regression–are considered the state-of-the-art for tabular data thanks to the precision of decision trees at the core of Random Forests methods. While single decision trees tend to overfit part of the parameter space, such as the Jupiter-sized planets, Random Forests leverage many decision trees and aggregates their predictions. This leads to much less bias (hence the absence of plateau) but potentially higher variability in predictions, especially in sparser areas of the parameter space. This can be see in the scatter for imputations at sizes in between the main super Earths and gas giant groups, and past $1{,}000\,\mathrm{M}_\oplus$.

Similar to section 4.6.1, GAIN produces a very poor imputation (lower left panel of Figure 4.8). Because GAIN does not directly use the observed values in the dataset, but rather tries to mimic them, heterogeneous and potentially inconsistent values end up being proposed for imputation, which leads to a large scatter.

The mass estimates obtained via the M-R relationship of Chen and Kipping [110] as computed by the Planetary Systems with Composite Parameters (PS-CP) show the overall higher error (see lower right panel of Figure 4.8). But this is mainly because the mass-radius piece-wise linear relationship is not invertible in the range $11.1\,\mathrm{R}_\mathrm{J}$ to $14.3\,\mathrm{R}_\mathrm{J}$, which despite being a small radius range, includes most Jovian worlds and spans over masses from 85 to $35{,}000\,\mathrm{M}_\oplus$, obviously leading to poor results in that range. For planets with masses below 85 $\mathrm{M}_\oplus$, the algorithm performs well and has one of the tightest estimates for the very small planets.

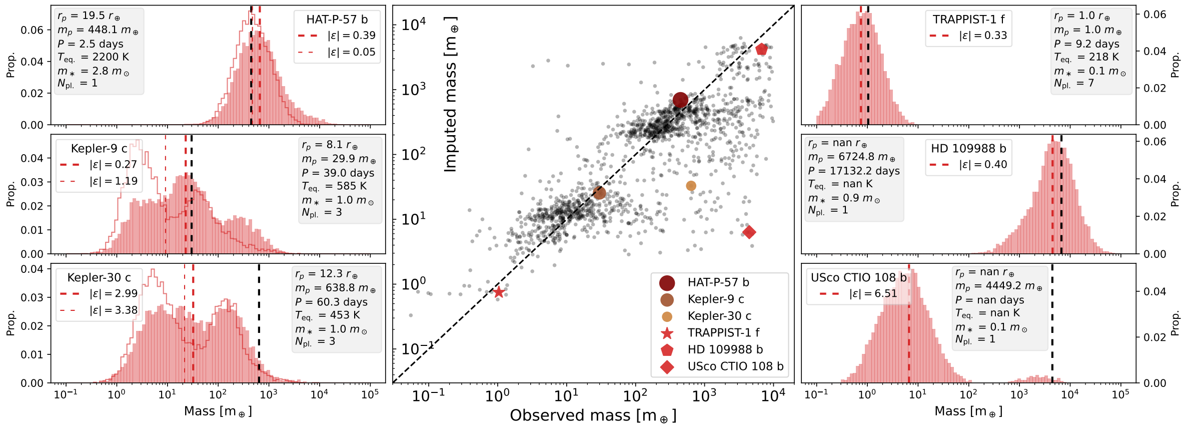


Figure 4.9: Six distributions for imputed mass values with the kNN×KDE in the “transit regime" where planet mass is concealed and imputed. The three profiles on the left are the same planets as in Figure 4.5, whose mass was originally imputed using the complete properties dataset. The distribution using the complete properties dataset is shown outlined in red (note that in some cases, the planet properties have been updated since TLG2020. For the correct comparison in this case, the red outline shows the distribution using the complete properties dataset for these updated values). The three profiles on the right were selected as interesting examples: TRAPPIST-1 f is an Earth-sized planet, and one of the smallest in the dataset, HD 109988b is one of the most massive planets in the dataset and does not have a radius observation, while USco CTIO 108b has one of the highest errors. The central plot is a reproduction of the kNN×KDE case in the top left panel of Figure 4.8, showing the location of the six planets.

Figure 4.9 looks at the mass probability distributions for six planets from Figure 4.8 with observed but concealed mass values that have been imputed by the kNN×KDE algorithm. The left-hand three panels show the mass distributions for the same planets as shown in Figure 4.5, but now imputed making use of the full archive dataset rather than the smaller complete properties dataset. Note that newer observations have been conducted between the development of this paper and TLG2020, resulting in a few planets having updated measured properties in the newer dataset utilized in this section. None of these changes have strongly altered the distribution shape for the three planet comparisons, HAT-P-57b, Kepler-9c and Kepler-30c. However, in order to accurately assess the impact of drawing on the full archive dataset for imputing properties, we recalculated the mass distribution for each planet with the updated measured values using the complete properties dataset in section 4.6.1. The updated distributions are shown as a red outline on the left-hand panels in Figure 4.9.

Interestingly, while there is an overall improvement in the error for the planets also present in the complete dataset, the error for HAT-P-57b has slightly increased. This is due to the addition of more massive planets present in the full dataset, a number of which orbit around higher mass stars and have higher equilibrium temperatures similar to HAT-P-57b. HAT-P-57 is a massive A-type star that is an unusual planet host in the current archive, and one where it would be easier to detect a high mass planet. The inclusion of these higher mass planets in the dataset therefore slightly extends the high mass tail in the imputed mass distribution for HAT-P-57b, giving a raised average mass value. However, the change is small with the observed mass value still lying at the peak of the imputed distribution.

By contrast, the mass prediction for Kepler-9c has substantially improved. While the previous distribution based on the smaller complete properties dataset had a bimodal shape that proposed two possibilities for the most likely planet mass, Figure 4.9 shows evidence of three modes, although less distinctly separated than in Figure 4.5. The middle of the three new modes has the highest peak, and lies at the observed value for Kepler-9c. To the left of the central mode is a lower mass option that peaks around $3\,\mathrm{M}_\oplus$. This mode lies in the same location as the stronger of the bimodal peaks in the previous distribution. On the right of the distribution sits a third high mass mode at about $1\,\mathrm{M}_\mathrm{J}$ that is at a slightly higher mass than the previous high mode peak but with much lower probability than the other two peaks.

The addition of the third more accurate mode in the mass distribution of Kepler-9c is the inclusion in the full archive dataset of more planets with observed mass and radius between the two most common classes of gas giant and super Earth. This can be seen by the addition of light gray dots on the planet mass versus planet radius plot in Figure 4.1. The broad profile still indicates that Kepler-9c is somewhat unusual, but the highest probability from the larger dataset does provide the correct solution.

The less distinct peaks for Kepler-9c push away from the concept of very distinct planet classes and point towards a continuum of planet sizes created by a multitude of evolutionary pathways during planet formation. This is seen again in the new distribution for Kepler-30c. The sharply peaked bimodal mass distribution in Figure 4.5 has softened to a broader spread of masses. The two original peaks are still visible at the same locations as in the previous distribution, but more options now sit in-between. However, in this case, the extra options have not improved the mass imputation which remains significantly lower than the observed mass. This is because the 60 day orbit remains relatively rare, even in the full archive, and even fewer planets at that orbit have high masses, as can be seen in the pairplot of orbital period and planet mass in Figure 4.1.

The right-hand column of Figure 4.9 shows three mass profiles for planets not studied in section 4.6.1. In the case of TRAPPIST-1f, the planet has a complete set of observed properties for the six considered parameters, but was in the training (not test) set for the mBM. The other two planets in that column have incomplete properties and so could not be included in the prior complete properties dataset.

TRAPPIST-1f is one of seven approximately Earth-sized planets orbiting a low mass M-dwarf star [140]. The planet orbits in the so-called habitable zone, which would allow surface liquid water to persist if the planet hosted an Earth-like environment [15]. Small worlds on temperate orbits are still relatively rare, as can be seen in Figure 4.1, as are planets around very low mass stars and in systems with more than six known worlds. The kNN×KDE therefore finds a fairly loose collection of neighboring planets, many of which orbit the more commonly observed G-type stars. Despite this, the neighbors are in systems of six, seven or eight planets and are consistently low mass. Six of the nearest and most highly weighted neighbors are unsurprisingly the other planets in the TRAPPIST-1 system, which have very similar sizes due to the “peas in a pod" effect discussed in section 4.3. The result is an accurate mass estimate, with the algorithm certainty demonstrated by a single, fairly narrow peak.

As can be seen in the central panel of Figure 4.9, planets with observed masses significantly below that of TRAPPIST-1f have overestimated masses. This is likely due to an absence of any close neighbours, forcing the kNN×KDE to look exclusively to higher mass planets for guidance. Weighting close neighbors assists the kNN×KDE with handling planets in a sparse area of the parameter space, but it inevitably struggles if the planet is close to unique amongst current observations. As previously seen, uncertainty in the imputation is evident in the broadness of the resulting distribution.

The last two mass imputations in Figure 4.9 are not technically a replication of a transit observations, as neither planet transits its host star and therefore have no observed radii measurements. Without an observed radius, the physical size of both HD 109988b and USco CTIO 108b therefore could not be estimated either by the previous mBM code in TLG2020, or by mass-radius relationships [e.g. 110, 113]. This therefore makes the planets an interesting test of the performance of the kNN×KDE algorithm.

The imputed mass for HD 109988b has a low error, with a distribution sharply peaked around the observed mass and a tail that extends more towards lower masses. With a mass over $21\,\mathrm{M_J}$, HD 109988b is more properly a wide-orbit brown dwarf [141]. A population of such celestial bodies with masses exceeding $1000\,\mathrm{M_\oplus}$ and orbital periods over 1000 days are present in the full archive dataset, but entirely absent in the complete properties dataset due to the the difficulty in measuring the full set of six properties. From the pairplot in Figure 4.1, HD 109988b can be seen to be typical of that distant, massive population of planets, explaining the successful imputation of its mass despite relatively few observed values. The low mass tail on the distribution is actually due to the planet's missing radius observation, which requires the kNN×KDE algorithm to search for neighbors with both measured planet radius and mass. Such coupling promotes consistency between all planet measurements, but reduces the potential neighbors in this planet group where most planets have been discovered by either the radial velocity or direct imaging methods, and do not transit. Relaxing that requirement might have tightened the profile in this case, but risked an inconsistent set of planet properties.

The final panel in Figure 4.9 shows the mass distribution for another very massive planet close to the deuterium burning limit, USco CTIO 108b. This planet was discovered through direct imaging and unlike HD 109988b, does not have a measurement for the orbital period. This leaves only the stellar mass and number of planets from which to impute a mass estimate. The stellar mass is also exceptionally low, significantly less than the majority of known host stars. As a result of the high number of missing values, and sparse parameter area around the known stellar mass, the resulting imputation is challenging. The twenty closest neighbors identified by kNN×KDE that have the required four missing properties are mainly situated in the super Earth region of Figure 4.1, due to smaller planets being slightly more commonly found around low mass stars. This creates the primary distribution peak at about $7\,\mathrm{M_\oplus}$, with only two neighbors indicating that the higher (correct) mass is possible.

The struggle with imputing properties for planets with a low number of measured values is not surprisingly, and evidence of this can also be seen in the central plot of Figure 4.9. A horizontal line of planets can be seen with imputed values all just below $10\,\mathrm{M_J}$ ($3000\,\mathrm{M_\oplus}$). These are planets all detected via gravitational microlensing, and similarly have only the mass of their host star and number of planets in the system from which to impute a mass value.

Despite the stronger peak at $7\,\mathrm{M_\oplus}$, a manual inspection of the mass profile for USco CTIO 108b would likely have resulted in the small high mass peak being consider the more likely value. This is because the algorithm does not use knowledge about the detection technique when estimating values. Such information was intentionally excluded to avoid the use of non-physical trends in the exoplanet demographics. However, in this case, the imaging detection points to a massive world. This is an additional example of where the distribution is more valuable than a single value imputation.

Mass and radius prediction in the RV regime: full archive dataset

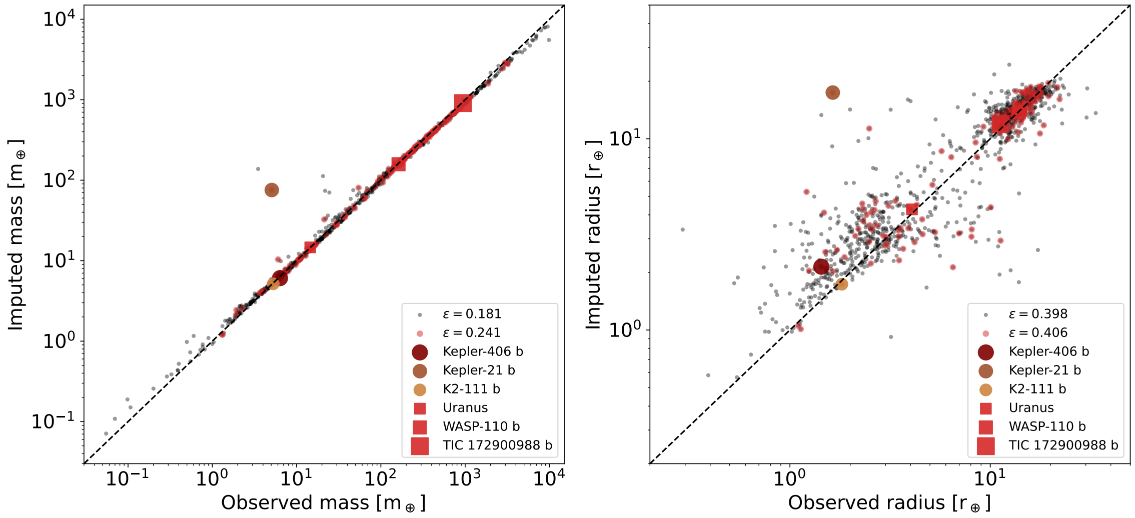


Figure 4.10: Test results for the kNN×KDE algorithm when using the full archive dataset and treating each planet as a radial velocity observation, with concealed and imputed planet radius and mass values, weighted by a given minimum mass measurement. The plots show the results for the mass imputation (left) and radius imputation (right), after the distribution has been weighted by the minimum mass. Black dots show all planets in the full archive dataset, while red dots show the planets that were also in the test set for the complete properties dataset. The profiles for the planets identified in the legend are shown in Figure 4.11.

Figure 4.10 now looks at the performance of the kNN×KDE algorithm in imputing both planet mass and radius based on the remaining four properties in the dataset, together with a minimum mass value, leveraging the full archive dataset. This is equivalent to treating each planet that has a measured mass and radius as if it were a radial velocity detection, and assessing code performance by imputing those (concealed) properties. As in section 4.6, this is a test that can only be performed with the kNN×KDE algorithm, since a distribution of imputed mass values is needed to convolve with a minimum mass measurement, as described in section 4.5.1. The black dots in Figure 4.10 show the results for imputing the mass and radius for the 1,081 planet in the full archive dataset with both measured values, while the red dots indicate planets that were also in the complete properties dataset of TLG2020.

As for the transit regime comparison in Figures 4.4 and 4.8, the average error for the planets common to both datasets has decreased with the use of the full archive dataset for the imputation. However, there is now an even lower error when averaged over all the planets in the full dataset, whereas in the transit regime test, the larger dataset had a higher average error. This error reduction in the radial velocity regime test reflects that the planets in Figure 4.10 have less variability in the data available for imputation, with all entries having a minimum mass measurement but no radius or true mass available. In the equivalent transit regime test in Figure 4.8, a planet radius measurement was available for part of the archive, resulting in a mix of imputations for planets with and without a size guide. Since the remaining four parameters (orbital period, equilibrium temperature, stellar mass and number of known planets in the system) are more commonly measured than planet radius, this variation in planet radius measurements for the transit regime results in more scatter. As with the radial velocity regime test for the complete properties dataset in Figure 4.10, the relation between the imputed mass and observed mass is very tight, due to the value of a minimum mass in guiding the true mass value. Notably, there is no averaged plateau for either the mass or radius imputed values. This is mostly due to the observed minimum mass value, whose convolution with the mass distribution prevents broad averaging over the full imputed distribution.

The distributions for the imputed masses and radii for particular cases highlighted in Figure 4.10 are shown in Figure 4.11. The top three profiles show the same planets as in Figure 4.7, but now with their properties imputed using the full archive rather than the smaller complete properties dataset. As with Figure 4.9, there have been new observations of the planets since TLG2020 dataset was made. The red outline therefore shows the distribution when the complete properties dataset is used to impute the mass and radius based on any updated properties. The average mass and radius values (the imputed values in Figure 4.11) are shown as red dashed lines, with the epsilon error in the legend. The thin dashed line is the result when using the complete properties dataset.

In contrast to the change in the profile shape seen in Figure 4.9 for the transit regime test, the use of the full archive dataset has a smaller effect on the profiles for Kepler-406b, Kepler-21b, and K2-111b. This is not very surprising. The kNN×KDE code performs the imputation of missing values by searching for neighbors in the planet parameter space that have measured values for all missing properties. In the case of the transit regime test, this often involves only requiring neighbors to have a measured mass value. The choice for potential neighbors therefore significantly expands when the full archive is utilized. However, in the radial velocity regime test, neighbors must always have both a mass and radius measurement. The neighbors involved in the imputation are therefore often also in the complete properties dataset (since planetary mass and radius have the highest missing rate as seen in Table 4.3), so there is a stronger overlap between the neighbor selection for the two datasets. This constraint is necessary to ensure that the imputed values are consistent values across properties, but does mean that it is harder to fully leverage a large

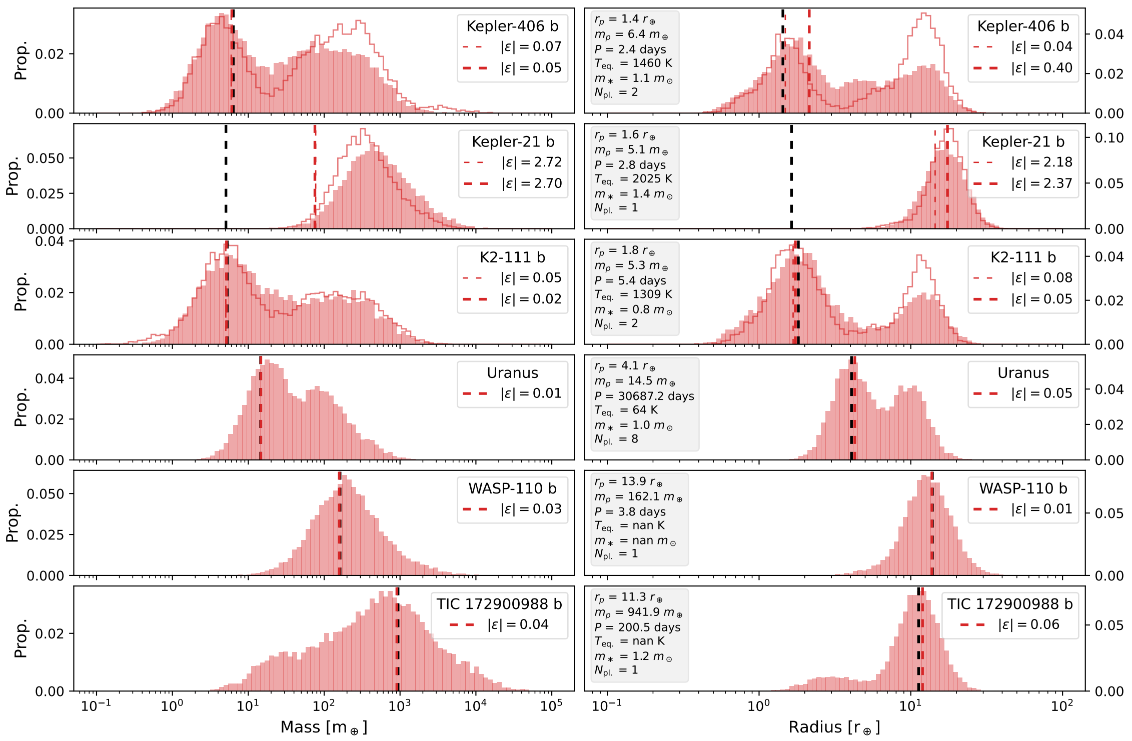


Figure 4.11: Distributions for the imputed mass and radius values calculated with kNN×KDE for the planets highlighted in Figure 4.10 using the full archive dataset. The top three planets, Kepler-406b, Kepler-21b and K2-111b, are the same planets whose distributions were shown in Figure 4.7 when imputed using the smaller complete properties dataset. The red outline shows the profile when calculated using the complete properties dataset, while the histogram is the profile with the full archive dataset. (As previously, the imputation with the complete properties dataset has been updated with the latest observations for the planet.) The average imputed value is shown by a red dashed line (thin dashed line for the complete properties dataset), with the error in the legend. The black dashed line is the observed mass and radius. The last three profiles are for planets that are not in the test set for the complete properties dataset.

incomplete database when imputing multiple values, as in the radial velocity regime. The kNN-Imputer algorithm does not have this constraint, but generally performs more poorly than the kNN×KDE, as seen in Figures 4.4 and 4.8. This issue will be returned to in section 4.6.3.

Despite the above restriction, the expansion to the full archive dataset has adjusted the distribution for Kepler-406b with a decrease in the probability of the higher mass and radius peak in the bimodal profile, and now more strongly favor a correct, smaller sized planet. When combined with the minimum mass distribution that also points to the smaller mass peak, this reduces the error for the average imputed mass value. The peak value of the radius distribution also agrees with the observed value, although the average value gives a higher error. The shift in the average is because the use of the full archive has flattened the higher mass peak and increased the probability of planets at intermediate sizes between the two peaks. While the probability of these planet sizes remains reasonably low, it is sufficient to push the average towards higher options. This change in profile shape is due to the wider range of radii values now observed for planets on short orbital periods with high effective temperatures, where heating can cause the atmosphere of planets to inflate [142]. In the case of Kepler-406b, the planet's average density is 11.8 gcm^{-3}, indicating a rocky planet without a thick atmosphere that would have the potential to cause variations in radius [143]. Consistent with this, the observed mass and radius for Kepler-406b lie near the main peak of the distribution at the low mass and small size end.

Conversely, the distributions for Kepler-21b show almost no difference between the imputations using the two datasets. The planet has a high error, as the observed value is substantially lower than that predicted by the code. In this case, the planet's very high equilibrium temperature and massive host star means that it still remains an outlier in the full planetary demographics. This would be indicated by a minimum mass measurement, which would require an orbital inclination of less than 1 degree to be consistent with the peak value.

The distribution for K2-111b is significantly changed from the profile discussed in Figure 4.7 due to the discovery of a second planet orbiting K2-111 since the original complete properties dataset was created. As discussed in section 4.6.1, this significantly increases the chances of a planet being smaller than a gas giant, and the distribution shape for K2-111b is now bimodal when either the complete properties or full archive dataset is used in the imputation, as seen by the red outline in Figure 4.11. While the mass profile shows only a small change when the full archive is employed, the correct smaller radius value is more strongly favored with the full archive. This is from the increase in smaller planets at high equilibrium temperature. The epsilon error for this planet is now extremely low, with the use of the full archive dataset offering the best match.

The next three profiles show planets that were not in the test set for the complete properties dataset. Our own Solar System's Uranus has a complete set of observed properties, but was in the training set for the mBM algorithm. Both WASP-110b nor TIC 172900988 b were only discovered recently in 2021, and neither has complete properties that would allow them to be included into the complete properties dataset.

As an outer planet in our own Solar System, Uranus is an outlier in its orbital period and equilibrium temperature. The majority of exoplanets found with periods longer than 10,000 days are usually young massive planets discovered by direct imaging. However, despite being quite an extreme outlier, the kNN×KDE algorithm accurately predicts the planet mass, with the observed value sitting close to the peak of the distribution, which is supported by the minimum mass, with the combined distributions placing the imputed value at the kNN×KDE peak. This is due to the presence of Neptune, which sits very close to Uranus across the parameter space and therefore becomes the most strongly weighted of the twenty neighbors used in the kNN×KDE imputation. The remaining neighbors span a range of masses, but are more weakly weighted. The gas giant population can be seen as a second, lower peak at high mass and radius in the distribution (driven mainly by Saturn and Jupiter), and smaller planets are selected due to our Solar System having high planet multiplicity (although very weakly weighted), which usually indicates smaller worlds.

Similar to TRAPPIST-1f, the accuracy of the imputed value for Uranus demonstrates the importance of using a weighted neighbor scheme when performing probability density estimation; planets in sparser areas of the parameter space can retain accurately estimated properties by favoring the small number of similar discoveries, rather than a non-weighted average which risks covering a large range of the parameter space. For planets in very sparse areas of the parameter space, it is worth noting that the imputation may depend on just two or three close neighbors.

WASP-110b belongs to the hot Jupiter population, with a mass between that of Jupiter and Saturn and an inflated radius larger than Jupiter [144]. There is no stellar mass nor equilibrium temperature recorded in the NASA Exoplanet Archive, so the imputed mass and radius values are based on the planet's orbital period and number of known planets in the system. However, the hot Jupiter population is densely clustered in single planet systems at orbits of a few days, allowing the kNN×KDE algorithm to make an accurately imputed mass and radius with a profile clearly focused on the gas giant population that is only slightly broader than that for hot Jupiter HAT-P-57b in Figure 4.9, where more data is available. The convolution with the minimum mass does not move the imputed value from the peak of the distribution.

TIC 172900988 b is a more complex case. The multi-Jupiter mass planet orbits in a binary star system with a circumbinary orbit that circles two stars of similar mass [145]. The actual mass of TIC 172900988 b is uncertain, with estimated values extending from $824\,\mathrm{M}_\oplus \leq M_p \leq 981\,\mathrm{M}_\oplus$, due to multiple solutions for the orbital properties. An analysis with an algorithm like the kNN×KDE presented here is a possible path to reducing the uncertainty, by an independent estimate of the planet properties based on similar planets in the multi-dimensional parameter space. This made the planet an interesting case study. However, its properties do also present challenges for the machine learning imputation. As there are very few binary systems in the NASA Exoplanet Archive, the stellar host number was not included as a property in the database that can be utilized by the algorithms. Moreover, the complexity of a temporally varying equilibrium temperature for the planet means that this parameter is additionally omitted. The stellar mass in this case was that of the primary star. Even with these challenges, the overall imputed values for the planet's mass and radius as seen in Figure 4.11 are a close match to the recorded observed value of $942\,\mathrm{M}_\oplus$ and $11.3\,\mathrm{R}_\oplus$, with the observed value lying close to the peak of the distribution, even without the minimum mass guide. The orbital period of TIC 172900988 b is long for the majority of planets in one planet systems, which is dominated by the close-in hot Jupiter population. The selected neighbors are therefore not as tightly gathered compared to WASP-110b, and have a broader range of values, with no very close neighbors dominating the weighting (differing from planets such as Uranus). This is reflected in the wide distribution width for the mass. The resultant distribution suggests that the recorded value is the most likely one for the planet.

4.6.3 The extended dataset: eight properties

The ability to utilize an incomplete database opens up the possibility to leverage information from more planet properties for the imputation of missing values. Previously, the database had been restricted to six properties that were selected to be informative about the nature of the planet, while also having sufficient observed values from which to build a training set of complete properties for the mBM network. The algorithms developed for this paper removes that restriction, and allows additional potentially informative properties to be included in the imputation of missing values, even where there is a high fraction of missing values in the archive.

In this section, the full archive database is extended to include the host star metallicity and the planet orbital eccentricity. As discussed in section 4.3, trends between pairs of variables indicate that both properties may assist the imputation of missing values. Note that as all properties are equally weighted in algorithms such as the kNN×KDE, adding uninformative properties to the imputation would risk lowering the accuracy of the imputation results. Both stellar metallicity and orbital eccentricity were missing from the previous datasets due to their low completeness in the NASA Exoplanet Archive (see Table 4.3). However, although non-complete datasets can be leveraged by the algorithms presented in this paper, a very low number of observed values does still present challenges. In particular, the kNN×KDE algorithm requires twenty neighbors in the parameter space to a planet whose properties are to be imputed, all of which must have observed values for all missing properties. If one or more property is rarely observed, the distance to the twenty neighbours can become very large, and the imputation proportionately poorer. This issue was mentioned in section 4.6.2, but now becomes a more serious problem for the algorithm due to the low completeness of our extended dataset. To tackle this issue, the kNN×KDE algorithm was adjusted so the user could choose whether to impute the value of all missing properties, or just a subset. For the properties which would not be imputed, the algorithm dropped the requirement that the selected neighbor planets had to have this property measured. This allows the low-completeness stellar metallicity and orbital eccentricity to be used to define which neighbors are closest within the eight dimensional parameter space, but accept neighbors for the imputation with only requested parameters for imputation. For consistency with the previous two sections, the kNN×KDE algorithm imputed all of the original six planet properties where missing, but did not impute the stellar metallicity or orbital eccentricity, using these exclusively to select neighbors with relevant properties.

We also drop the other four algorithms at this stage in the analysis. The kNN×KDE algorithm has been a top performer for the previous two datasets, and the information from the probability distributions that the kNN×KDE can create greatly exceeds that available from a single imputed value.

Mass prediction from transit observations (extended dataset)

Figure 4.12 shows the results from the kNN×KDE algorithm for the transit regime test, where known mass values are concealed and estimated as might be required for imputing missing mass values in transit observations. This is the same test that was performed for the full archive dataset in Figure 4.8, but now with eight properties including the planet orbital eccentricity and stellar metallicity used in the imputation. Note that the orbital eccentricity is included in the imputation where present, even if this would not normally be measured as part of a transit observation. Likewise, not all the planets plotted above have radius measurements. The epsilon error over the full dataset has decreased slightly with the addition of two extra parameters, moving from $\epsilon = 1.510$ (six parameter dataset) to $\epsilon = 1.502$ (eight parameter dataset), demonstrating a small overall improvement when

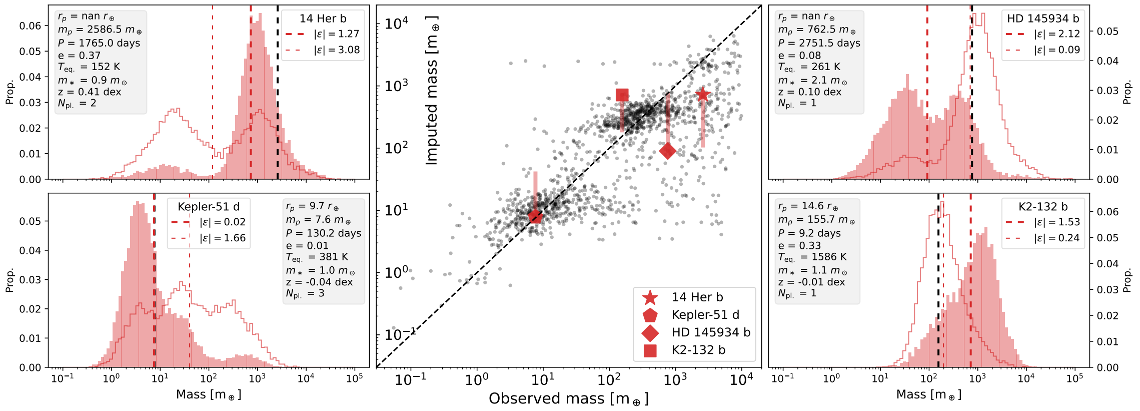


Figure 4.12: Test results when using the extended archive which leverages eight planet properties during the imputation. This is the transit regime test, where known planet mass values are concealed and imputed by the kNN×KDE algorithm, similar to what would be needed to impute mass for a transit observation. The ϵ error for the 1,426 planets plotted is $\epsilon = 1.502$, and when computed only over the planets that were in the test dataset of TLG2020, the error is $\epsilon = 0.840$. This demonstrates a small overall improvement in mass imputation accuracy. The two planet profiles on the left show planets where the error decreased with the inclusion of two extra planet properties in the imputation. On the right, are two planets where the error degraded with the extra information. The red outline on each histogram is the distribution for the planet when using the previous six property dataset. The filled histogram is the distribution when using the eight property dataset. The location of these planets on the central mass imputation plot are shown with a red line indicating their original imputed value when using the previous six-property full archive dataset.

adding additional information. A similarly small improvement can be seen on the subset of planets in the test set of TLG2020, going from $\epsilon = 0.846$ (six parameter dataset) to $\epsilon = 0.840$ (eight parameter dataset).

To take a closer look at how the extra information in the extended dataset can impact the planet property imputation, four planets are highlighted in the central plot in Figure 4.12 that have measured values for orbital eccentricity and stellar metallicity. The imputed mass value using the eight properties extended dataset is marked with the red shape, and a red trail indicates the location of the previous imputed value when utilising the six properties dataset. The profiles for each planet are shown flanking the central plot. 14 Her b and Kepler-51d shown on the left side of Figure 4.12 have greatly improved imputed mass values, whereas the imputed mass values for HD 145934 b and K2-132b deteriorated with the additional information.

14 Her b (also known as HD 145675 b) is a massive gas giant on a Jupiter-like orbit with a period of 4.8 years around a Sun-like star [141, 146]. Discovered via radial velocity, this planet does not have a radius measurement, so its mass during this transit regime test

is based on orbital period, equilibrium temperature, stellar mass and number of planets when using the six properties full archive dataset, and these four properties plus orbital eccentricity and stellar metallicity when using the new extended properties dataset. In the pairplot shown in Figure 4.1, 14 Her b belongs to the cluster of high mass, long period planets that can be seen towards the top right in the planet mass versus orbital period plot. Notably, this group of planets rarely transit, so very few have radius measurements. This means that the kNN×KDE algorithm has to go outside this cluster for neighbors with measured values for the six main properties, stepping into the parameter space for the longer period gas giants down to super Earths. The result is a bimodal distribution when utilising the six properties full archive dataset, with peaks close to the measured gas giant value and at the super Earth mass of about $11\,\mathrm{M}_\oplus$. With the inclusion of the orbital eccentricity and stellar metallicity, this degeneracy breaks, and the higher gas giant mass is strongly favoured by the kNN×KDE algorithm. This is because the relatively high orbital eccentricity at $e = 0.37$ is far more commonly found for high mass planets; a trend that was noticed in the discussion of the Pearson correlation coefficients in Figure 4.2. Removing or lowering the weighting on planets with lower eccentricity measurements for the close neighbors therefore increases the algorithm's uncertainty that this is a gas giant.

Kepler-51d is a second case where the addition of the orbital eccentricity and stellar metallicity has greatly increased the certainty of the kNN×KDE algorithm to favour a particular planet mass regime. In this case, the profile has changed from a fairly continuous distribution between a gas giant and rocky planet, to a strongly peaked profile at a few Earth masses. Kepler-51d is an unusual planet in the archive as it has a very low density [147]. Based on a radius measurement alone, a gas giant would be suspected. With the six properties available from the full archive dataset imputation, the multi-planet system and orbital period suggest lower masses may be equally probable. However, both the orbital eccentricity and stellar metallicity offer clues to narrow down the distribution. The orbital eccentricity for Kepler-51d is very low, which is common for a wide range of planet masses. However, many high mass planets with low eccentricity will probably be in tidal lock, on much shorter orbital periods than Kepler-51d. Moreover, Kepler-51 has a sub-solar metallicity, which slightly favors lower mass planets. This adjusts and re-weights the neighboring planets so that the lower mass becomes the strongly dominant peak.

On the right-hand side of Figure 4.12, two profiles are shown for planets whose mass imputation significantly deteriorated with the additional information from orbital eccentricity and stellar metallicity. The profile for HD 145934 b moved from a strong (and correct) estimation that this was a gas giant, to an equal chance of both a gas giant and super Earth. Like 14 Her b, HD 145934 b belongs to the group of long period, high mass planets discovered via radial velocity that do not have radius measurements. However, the high mass of the star HD 145934 and the lack of a second planet in the system means that the six properties full archive dataset is confident that this is a gas giant. This certainty is likely upset by the low orbital eccentricity of HD 145934 b when the imputation is based on the eight properties extended dataset. Low eccentricity is expected for single planet systems with less scattering and also for planets with lower masses, as is seen in the data (see Figure 4.2). The low eccentricity value therefore increases the parameter space distance from high mass neighbors that also have high eccentricity, and creates a more even probability between the two mass options.

The last of the four distributions in Figure 4.12 is K2-132b. The addition of orbital eccentricity and stellar metallicity has pushed the planet distribution to higher masses, taking the peak further from the measured value. K2-132b is an inflated gas giant that sits in a dense region of the six properties full archive parameter space. This produces a single, steep peak very close to the measured mass. However, the planet has a high eccentricity which is more unusual for a large planet on a short orbit. The star evolution has been proposed as a reason for this high eccentricity, with tides on evolved stars causing a transient high eccentricity orbit for the host planet [148]. The inclusion of the eccentricity therefore selects more highly eccentric neighbors, which favours more massive planets. The result is a quite broad profile (indicating some uncertainty in the imputation) that is skewed towards higher masses. This might be avoided by including spectral type in the imputation, but the value is only recorded in the archive for a relatively small number of entries. K2-132 b is therefore actually an example of an outlier planet, which only appeared to be typical when considering a reduced number of properties.

Mass and radius prediction in the RV regime: extended dataset

As with the previous two datasets, we can extend the imputation to estimating both the planet mass and planet radius when utilizing a measured minimum mass value, as would be common with a radial velocity observation. The result of imputing known masses and radii values in the exoplanet archive with the eight parameter extended dataset is shown in Figure 4.13. The error for the planet mass averaged over the 1,081 planets plotted after convolution with the minimum mass has lowered with the inclusion of the additional two planet properties from $\varepsilon = 0.181$ (six parameter dataset) to $\varepsilon = 0.157$ (eight parameter dataset), with the average error on the imputed radius also dropping from $\varepsilon = 0.398$ (six parameters) to $\varepsilon = 0.363$ (eight parameters). The imputation slightly worsens if we consider the planets in the original TLG2020 test set, changing from $\varepsilon = 0.241$ (mass) and $\varepsilon = 0.406$ (radius) (six parameters) to $\varepsilon = 0.275$ (mass) and $\varepsilon = 0.396$ (radius) (eight parameters).

Below the plots of the average imputed values, mass and radius distributions for three planets are shown that demonstrate a change in the profile shape due to the addition of the orbital eccentricity and stellar metallicity properties. Kepler-98b is a super Earth whose high error is substantially reduced by the inclusion of the additional two properties in the imputation. Using the six property full archive dataset, the kNN×KDE algorithm believes the planet to be a Jupiter-sized gas giant based on the planet's orbital period, equilibrium temperature, host star mass and number of planets in the system. This is principally driven by the 1.5 day orbital period and single planet status, which is very common in the hot Jupiter population. The nearest neighbors to Kepler-98b therefore end up being universally gas giants. The orbital eccentricity is not recorded for Kepler-98b, so the eight parameter database leverages only the stellar metallicity. While higher metallicity stars are more likely to host gas giants that stars of lower metallicity, smaller planets are also commonly found in orbit. This appears to be enough to slightly expand the knot of nearest neighbours away from the exclusively hot Jupiter population and produce a second small peak at lower planet mass in the distribution. In combination with the minimum mass measurement that indicates a lower mass planet, the final mass estimate is closer to the measured value. In this case therefore, the added planet property

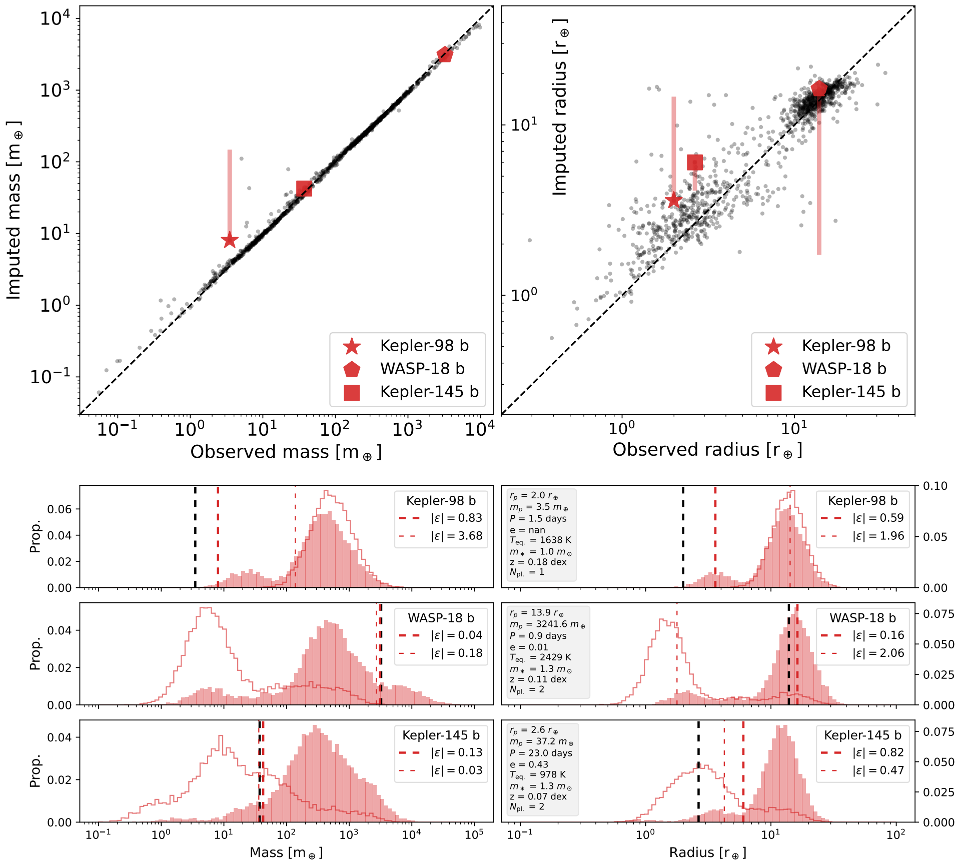


Figure 4.13: Test results when using the extended archive with eight planet properties for imputing both planet mass and radius with the inclusion of a minimum mass value "radial velocity" with measured planet mass and radius values. The average error for the mass imputation is $\varepsilon = 0.157$, and the radius imputation has an average error of $\varepsilon = 0.363$, showing a small improvement in accuracy compared with the previous six properties full archive data set. The imputation is explored further in three mass and radius distributions shown in the lower half of the figure. The filled histogram shows the property distribution using the eight parameter extended dataset, while the red outline shows the distribution when imputed with the previous six properties full archive dataset. The average imputed value is shown by a thick red dashed line for the eight properties extended dataset and thin dash line for the six properties dataset. The error for both is shown in the legend. The black dashed line is the measured value. The planet measured properties are shown in the gray box. The location of these planets is also marked on the upper panels, with a red trail indicating the previous imputed result with the six properties full archive dataset.

has helped to diversify the nearest neighbors.

Unlike Kepler-98b, WASP-18b actually is a hot Jupiter and it seems initially surprising that the planet would have a high error on its mass and radius with any chosen dataset. However, WASP-18b is a particularly massive planet at $10\,\mathrm{M_J}$, on an orbit of less than 1 day. The planet is so close to its host star that tidal interactions are likely on the brink of destroying the planet [149]. With no mass or radius measurement to initially guide the kNN×KDE imputation, the ultra-short orbit and high equilibrium temperature, coupled with a second discovered planet in the system, suggests a rocky or super Earth size for the planet. Adding in the orbital eccentricity and stellar metallicity in the extended dataset significantly improves this imputation, reflecting the peaks of the distribution so that a high mass planet is favored. It is not immediately obvious why this would occur, as the tidal circulation of orbits means that low eccentricities is expected for both high and low mass planets with short periods. However, the exceedingly small eccentricity for WASP-18b, and the higher metallicity of the star, has increased the proximity of a group of low eccentricity, higher mass planets in the parameter space that can be seen in the top left corner of the planet radius vs. orbital eccentricity pairplot in Figure 4.1, causing these gas giants to be favored as close neighbors.

As all properties are weighted equally in the kNN×KDE algorithm, the use of eight properties also lowers the significance of each individual property. This may have been advantageous in the case of Kepler-98b and WASP-18b, as it lowers the significance of the number of planets in the system which was misleading in these cases.

Unlike the previous two planets, Kepler-145b is an example where the imputed planet mass and radius has degraded as a result of the extra properties in the extended dataset. Kepler-145b is a planet between Neptune and Saturn in size, around a massive F-type star on a 23 day orbit. When using the six properties full archive dataset, the kNN×KDE algorithm favors a planet of the correct radius but lower mass. This is not surprising when looking at the planet's location on the planet mass vs. radius pairplot in Figure 4.1, as the planet has a high density. Discovered by transit timing variations, there is a large error in the observed mass estimate, so it is possible that the recorded observed mass is an overestimate [150]. When the dataset is expanded to eight properties, the imputed distribution peaks at a higher mass and radius than observed. This is driven by the high orbital eccentricity of the planet, which indicates a high mass world. It is maybe worth noting that the measured eccentricity is also in doubt, as Van Eylen and Albrecht [151] who reported the measurement noted that the transits of Kepler-145b were too shallow for any meaningful constraints on eccentricity. The outer planet, Kepler-145c, is thought to be in a close to circular orbit, so it could be that the eccentricity for Kepler-145b is artificially high in this instance.

4.7 The exoplanet distribution

In addition to imputing missing values in a dataset, the kNN×KDE algorithm can be used as a fully generative model to create an arbitrary large number of planets with entirely synthetic properties that still maintain the statistics of the observed population dataset. This synthetic planet population will have a complete set of properties and can therefore be used to explore the planet demographics using algorithms to visualise clusters within the high-dimensional parameter space. While such statistically defined clusters have no guarantee a corresponding physical meaning, close groups of planets within the parameter space may indicate shared evolutionary pathways, and additionally shed light on the underlying workings of imputation algorithms such as the kNN×KDE. Such an analysis would usually not be possible with an incomplete dataset that includes missing values.

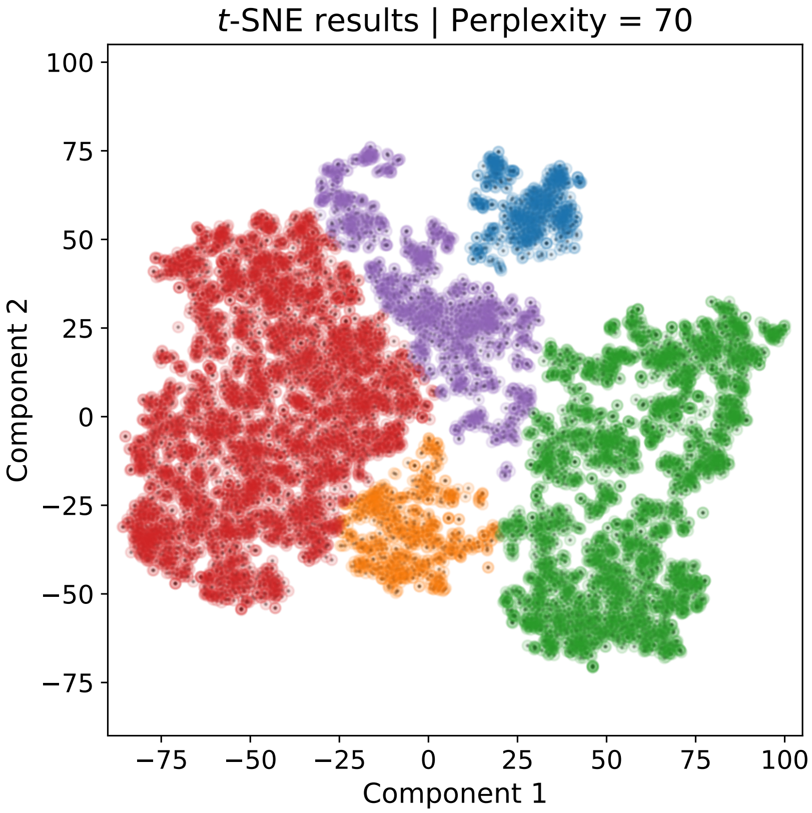


Figure 4.14: A 2D visualization of clusters of planets within the six dimensional parameter space, created using the t-SNE algorithm on a population of 10,000 simulated planets generated by the kNN×KDE. Five clusters identified by eye have been color-coded for subsequent analysis. The proportion of the planet population in each cluster is 5.5 % blue, 6.9 % orange, 32.5 % green, 42.7 % red, and 12.4 % purple.

Based on the extended exoplanet archive with eight properties, 10,000 planets have been generated with the kNN×KDE. These synthetic planets are then passed to a t-SNE algorithm in order to perform a visual clustering. The last variable corresponding to the number of planets in the system has been purposefully discarded before applying the t-SNE, because this variable not only takes discreet values (which can bias the clustering results), but also because the number of planets in a system does not carry much physical significance and is likely to be an observational bias.

Based on the eight property extended planet dataset, the kNN×KDE was used to create a synthetic population of 10,000 planets. This large new, full properties, synthetic population was analyzed by a t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm; a statistical technique that gives each datapoint in a multi-dimensional parameter space a location on a two-dimensional map that can then easily be visualized [152]. For this analysis, the eight properties extended dataset was used, but the number of known planets in the system was discarded before passing the data to the t-SNE. This was because this last variable only takes discreet values and it was found to strongly dictate the resulting clusters. Moreover, since this property is not necessarily correct due to undiscovered planets, it carries more observational bias and less physical significance than the other seven planet properties considered in this study.

The t-distributed Stochastic Neighbor Embedding (t-SNE) statistical method consists in projecting high dimensional data (7-d in this case, because the number of planets in the system is not considered here) into the 2-d plane for visualization purposes [152]. This method uses non-linear transformations in such a way that nearby points in the 2-d plane are also in the same vicinity in the original parameter space. Results of the t-SNE embedding are shown in Figure 4.14. Despite satisfying visual results after t-SNE embedding, further investigations should be performed to understand the origins of the apparent clusters. For the rest of this section, five color-coded clusters shown in Figure 4.14 have been selected. At this point, it is worth underlying that these clusters are arbitrary, and nothing prevents from finer or coarser clusters to be chosen instead. The selected clusters include two dominant groups and three smaller groups. The red and green clusters account for 42.7 % and 32.5 % of planets respectively and are well separated. The blue, purple, and orange clusters lie in the boundary between the two major clusters, and they account for 5.5 %, 12.4 %, and 6.9 % of planets respectively.

In projecting the 7D data on a 2D plane, the t-SNE uses non-linear transformation such that nearby points in the 2D plane are also in the same vicinity within the original parameter space. The result of the t-SNE is shown in Figure 4.14. Five clusters were identified by eye in the 2D plane, and manually colored in Figure 4.14. It is important to note that the choice of the five clusters is arbitrary, and nothing prevents the selection of finer or coarser clusters. The selection made here includes two dominant groups and three smaller groups. The red and green clusters account for 42.7 % and 32.5 % of 10,000 synthetic planets respectively, and are well separated in the 2D plane. The blue, purple, and orange clusters lie in the boundary between the two major clusters, and account for 5.5 %, 12.4 %, and 6.9 % of planets respectively.

To further characterize the selected clusters, a Principal Component Analysis (PCA) decomposition has been performed. PCA is another dimensionality reduction technique to embed high dimensional data into a smaller dimensional space. However, unlike t-SNE which is non-linear, the PCA linearly transforms the original data into a new coordinate system, therefore enabling for easier interpretation, although at the cost of less flexible embeddings. Additional figures for the PCA results are available a online additional material

The axes of the new coordinate system are ranked by explained variance decreasing order, such that only the first relevant components are used to analyze the embedded data. Figure 4.15 presents the PCA results for the 10,000 synthetic planets using the first four principal components, which respectively explain 44.6 %, 21.9 %, 13.0 %, and 10.3 % of the total variance. The breakdown of the principal components, also known as eigenvectors, are presented in Table 4.16. The applied color coding scheme follows the chosen clustering of Figure 4.14.

Finally, Figure 4.17 is a pairplot of the 10,000 generated planets, similar to Figure 4.1,

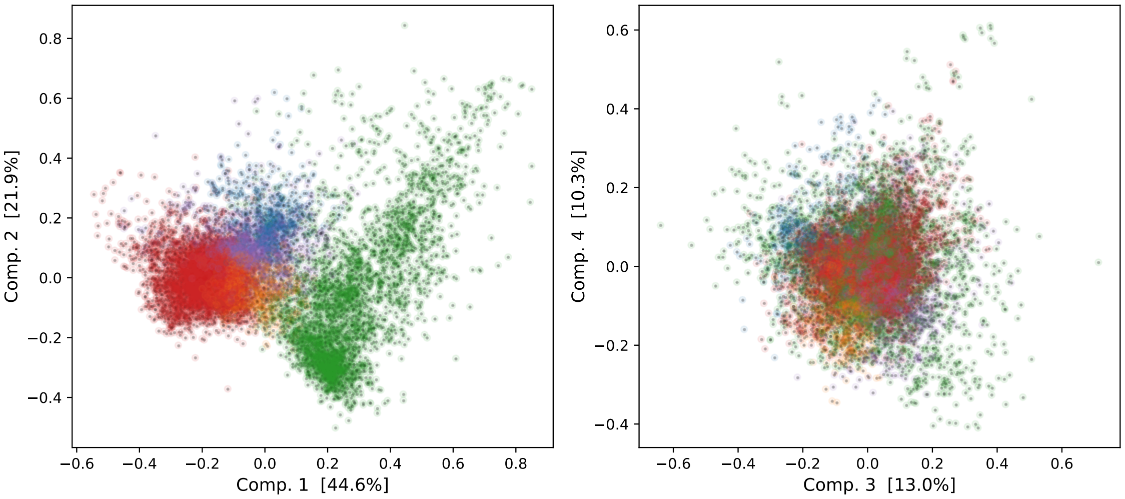


Figure 4.15: Visualization of the four principal components of the PCA results for the 10,000 synthetic planets. The color scheme refers to the clusters selected in Figure 4.14. The percentage of explained variance by each principal component is indicated alongside their corresponding axis.

Property	PC1	PC2	PC3	PC4
Planet radius	0.61	-0.35	0.20	0.03
Planet mass	0.64	-0.18	0.10	0.15
Planet orbital period	0.19	0.31	0.34	-0.21
Planet orbital eccentricity	0.38	0.67	-0.56	0.25
Planet equilibrium temperature	-0.08	-0.51	-0.54	0.39
Host star mass	0.06	-0.09	-0.10	-0.03
Host star metallicity	0.16	-0.16	-0.47	-0.85

Figure 4.16: Eigenvectors for the first 4 principal components of the PCA. Corresponding embedded data is presented in Figure 4.15.

where the colored clusters again correspond to the chosen groups after the t-SNE clustering. Note that we also apply the color scheme to the last variable corresponding to the number of planets in the system, but this variable has not been used for the t-SNE and the PCA results. Because the 10,000 synthetic planets do not have missing data, the univariate distributions in the diagonal of the pairplot Figure 4.17 are more faithful than on the pairplot Figure 4.1. Most notably, the histograms for planet masses and planet radii for this new pairplot now better reflect the actual distributions for the whole Exoplanet archive. Indeed, the radius histogram of Figure 4.1 shows a large peak for Super Earths and Mini Neptunes, because most planets with a measured radius have been detected via the transit method. But as these planets do not have a measure mass, they are lacking in the mass histogram of Figure 4.1, which instead shows a strong bias towards Jupiter sized planets. In other words, the histograms in Figure 4.1 have different numbers of observed planets for each property, but this is not the case anymore with the new pairplot of Figure 4.17 that has exactly 10,000 planets for each histogram and each 2d scatter plot.

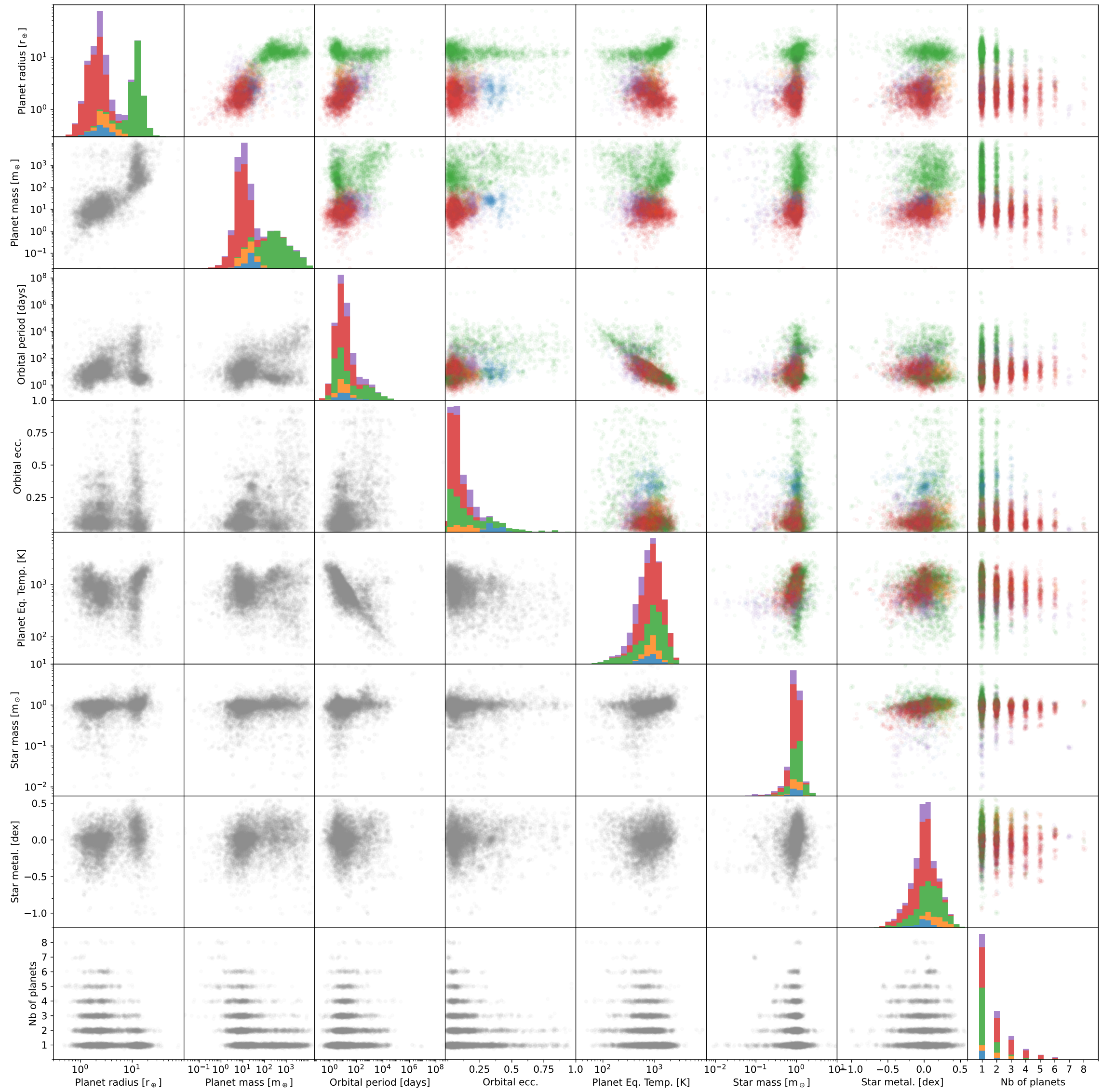


Figure 4.17: Pairplot for the 10,000 synthetic planets generated by the kNN×KDE. The color scheme refers to the clusters selected in Figure 4.14.

Upon analysis, the most dominant red group in Figure 4.14 corresponds to planets with small radius, low mass, short orbital period, and circular orbits. These are typical Super Earths which sometimes are part of multi-planetary systems (typical planet: Kepler-338e). The green group is the second biggest group in Figure 4.14, diametrically opposed to the red group. This green group is composed of large and heavy planets, at all range of orbital period and with potentially very elliptical orbits: these are the gas giants (typical planet: bet Pic b). Next, the purple and the orange groups are lying at the transition between the red and the green groups. These two groups overlap in Neptune-sized regime, with the purple group having longer orbital period, lower planet equilibrium temperature, lower star mass, and lower star metallicity (typical planet: K2-266 d), while the orange group is more compact and shows planets with shorter orbital period, high planet equilibrium temperature, and heavier host stars with much higher metallicity (typical planet: Kepler-94 b). The purple and orange clusters could alternatively have been grouped together, or merged with the red group of Super Earths. At last, the blue group lies close to the purple group while being well separated and very compact. It includes planets having particularly short and eccentric orbits, as well as having high mass for their small radius (typical planet: HD 18599 b).

Analysis of the clusters in Figure 4.14 reveals that the largest red cluster corresponds to the majority of the super Earths. Red cluster planets have radii below about $6\,\mathrm{R}_\oplus$, short orbital periods and circular orbits. In Figure 4.17, they largely fill the lower left planet population in the planet mass vs. radius plot. The two clusters diametrically opposite in Figure 4.14 colored green and yellow are the gas giants. Both clusters contain large and massive planets but are distinguished by their orbital period and eccentricity. The upper yellow cluster have orbital periods longer than 50 days and a wide range of eccentricities. The lower more compact cluster all have short periods and low eccentricity. These green cluster members are the hot Jupiters, and the density of the cluster reflects a fairly uniform set of properties. The close proximity to their long orbit counterparts could be support the main formation theory that hot Jupiters form through the same mechanism as cooler gas giants and migrate inwards. However, the dataset does not contain information such as composition, which would be one of best indicators of a different formation mechanism.

Finally, the small, compact blue cluster is one of the most clearly defined clusters in Figure 4.14. Blue cluster planets have universally short and eccentric orbits at super Earth masses. In the pairplot in Figure 4.1, they represent the extension towards high eccentricity from the super Earth planet sizes on the planet radius vs orbital eccentricity plot. In Figure 4.1, this group is present but not as clearly defined as the high eccentricity gas giants. But the generated synthetic planet population increases it prominence. The planets in the blue cluster also have quite high masses for their radii, giving them above average densities. These may be planets undergoing dynamical evolution, due to interaction with another planet in the system or previous scattering event. If so, the group properties do differ from other planets as the system is not yet settled.

The t-SNE provides a way of classifying groups of planets based on their properties in a multidimensional space. However, while some distinct clusters do exist, the properties these correspond to are clearly not unambiguous. This is similar to the shape of the probability distribution found by the kNN×KDE, which often indicated the presence of continuous range of possible values, rather than extremely distinct options. Planet classes are therefore likely also to be a continuous scale, without sharp distinctions.

4.8 Discussion and Conclusions

The NASA exoplanet archive is an invaluable source of data on the measured properties of the known extrasolar planets, providing the ground truth about what we know regarding the planet population. However, information on the planet demographics is challenging to extract from the archive because the dataset itself is incomplete, and because planet formation and evolution depends on a multitude of interconnected factors.

4.8.1 Imputing missing values

This paper explores three different ways to use machine learning to mine information from the exoplanet archive. The first compares the performance of five different algorithms to impute missing planet properties based on the measured properties of the planet population recorded in the archive: the kNN-Imputer, MICE, MissForest, GAIN and the newly developed kNN×KDE (see section 4.5). Unlike previous value estimates such as in TLG2020, all five codes could utilize incomplete datasets where each entry had only a subset of possible properties measured. This allowed the measured properties of all known exoplanets to be utilized in the calculation of the imputed value, independent of attributes such as discovery technique. Four of the algorithms calculated a single point estimate of the imputed value, while the kNN×KDE returned a probability distribution that was averaged for the algorithm comparison.

Two different datasets were compared for imputing missing values. The first of these was the “complete properties dataset" which included 550 planets all of which had six measured values for the properties planet mass, planet radius, orbital period, effective temperature, stellar mass and number of known planets in the system. The second was the “full archive dataset" which an incomplete set of those six properties for all 5,251 planets in the exoplanet archive. The imputation focused primarily on mass, as this is one of the most difficult planet properties to measure and has a high missing rate in the archive (see Table 4.3).

When utilising the complete properties dataset, the overall performance of all five codes was comparable, and performed slightly better than the modified Boltzmann Machine (mBM) neural network presented in TLG2020. The average error when imputing the planet mass for a test set of 100 planets was between 0.88 - 0.97, corresponding to an imputed mass within a factor of 2.4 - 2.6 of the observed value. The exception was the GAIN algorithm, which consistently gave the worst performance due to a “mode-collapse" problem where the large number high mass gas giants in the dataset caused an overestimation of the mass throughout the planet population.

When the dataset was expanded to the full archive dataset with incomplete values, the average error on the mass imputation for the same test set reduced for three of the algorithms, the kNN×KDE, MissForest, and MICE. The error range for these algorithms decreased to 0.83 - 0.92 (a factor of 2.3 - 2.5 of the observed value), marking a slight improvement. This was the hoped for result, as the full archive contains a factor of ten more planets and so should provide more information to increase the accuracy of the imputation. The fact that the improvement was not greater is due to the increase in range of the full archive dataset. A wider range of planet properties allows planets whose properties lie outside those within the complete properties dataset to be imputed, but does not increase the density in all areas of the parameter space which would result in a lower error.

However, the error when moving to the full archive dataset increased for the kNN-Imputer and GAIN. For the kNN-Imputer, this degradation was due to a strong bias towards two average masses values for the gas giants and super Earths populations. Similar but less extreme biases were also seen for MICE and kNN×KDE for the high mass planets. For the kNN×KDE algorithm, the bias came from averaging over the probability distribution. This could be reduced by lowering the number of neighbors (a code hyperparameter) used by the algorithm, at the cost of a less informative distribution. No bias development was seen for MissForest, while the scatter for the GAIN became too extreme to confirm the presence of any averaging.

Based on this, the two most recommended algorithms for the imputation of missing values would therefore be the kNN×KDE and the MissForest. The MissForest scheme uses a Random Forest for regression which has previously been lauded for tabular data, and shows no bias development due to the presence of dominant populations. However, the ability of the kNN×KDE to return a probability distribution is considered the most useful. The distribution can be combined with other sources of information to achieve the most accurate imputation. The most obvious use case is the inclusion of a minimum planet mass that is returned for detections made with the radial velocity detection technique. The minimum mass provides a second distribution of probable masses for a planet that can be combined with the kNN×KDE distribution to return an estimated mass value. In this case, the average error on the mass imputation reduces to 0.29 (within a factor of 1.3 of the observed value) for the complete properties dataset and 0.18 (a factor of 1.2) for the full archive dataset. Peaks in the distribution can also be manually selected for the imputed value rather than taking an average. This prevents the bias generated by a low probability distribution tail, or allows for other considerations (such as detection technique) to override the algorithmic most probable value, as discussed for USco CTIO 108b in section 4.6.2.

4.8.2 The kNN×KDE imputation distribution

In addition to allowing the inclusion of additional factors in the imputation, the probability distribution returned by the kNN×KDE can be used to understand the origin of the imputed value. This insight can be used not only to judge the accuracy of the imputation itself, but can reveal information such as whether the planet properties are common or rare amongst the known exoplanet demographics.

The probability distribution is created by weighting the properties of neighboring planets within the six or eight property parameter space. Broadly speaking, a probability distribution that has a wide peak, or flat profile, indicates that the planet's known properties are not sufficient to strongly favor a particular value, and that the imputed value may have a high error. Conversely, a more certain prediction will form a peaked distribution. A strong peak will often indicate that the planet belongs of a dense population of planets with similar properties, such as the hot Jupiter HAT-P-57 b in section 4.6.1 and section 4.6.2. Similarly, multiple peaks in a distribution can indicate that two possible values for the imputed planet property are consistent with known values.

However, more information can be gained by examining the properties of the neighboring planets identified by the kNN×KDE. A method for visualising this is to mark the neighboring planets on the pairplot in Figure 4.1, which can show the distribution of their properties. Examples of this plot for the planet profiles considered in this paper are shown in the supplementary online material. In the case of TRAPPIST-1f in section 4.6.2 and Uranus in section 4.6.2, a relatively peaked profile was formed due to the presence of a few very close neighbors. While these may be good indicators of the planet property (as in these cases), it is a small number estimate and not reflecting a large population group. Similarly, the bimodality for Kepler-30c in section 4.6.1 is due to the planet's rarity within the parameter space. The resulting neighbors are therefore quite dispersed, covering multiple masses groups.

Using the probability distribution profile in conjunction with other information can also indicate a high or low level of accuracy. As seen in sections 4.6.1, 4.6.2 and 4.6.3, the kNN×KDE distribution can be convolved with the distribution of possible masses from a minimum mass measurement. This can help narrow down uncertainty in the case of multiple distribution peaks, such as for Kepler-406b in section 4.6.1. A minimum mass value that lies far from the main peaks in the distribution can also indicate when an imputed value would have a high error. This occurred for K2-111b in section 4.6.1, where the poor performance by the kNN×KDE at estimating the planet's measured mass was indicated by a minimum mass that was outside the range of the distribution. In the case of K2-111b, this mismatch flagged the presence of a second, undiscovered planet in the system that was detected after the complete properties database had been created. The probability distribution with this planet included is shown in Figure 4.11, and agrees well with the minimum mass. Imputing the distribution of measured properties can therefore also reveal more about the planet, and could be a useful tool in target selection for follow-up missions.

The addition of more information in the dataset, both in increasing the number of planets between the complete properties dataset and the full archive dataset, and in adding two more properties in the extended properties dataset, generally improved the imputation results with the kNN×KDE. The change in shape of the probability distributions also indicated a shift in the underlying demographics. Peaks in the distributions in the full archive were typically less sharply defined, such as the mass distribution for Kepler-9c and Kepler-30c in section 4.6.2. This was indicative of more planets being found an intermediate sizes, and pointed to a more continuous range of properties for planets, rather than distinct classes.

The ability to dissect the probability distribution by examining the neighbors is a strength of the kNN×KDE. Used in conjunction with 2D plots such as the pairplot in Figure 4.1, the distribution structure can be easily understood (although it would be a more difficult task to do this in reverse and guess the distribution for a planet based on the 2D plots, due to the multidimensional dependencies). This is an advantage a statistics-driven algorithm has over move opaque schemes such as the mBM in TLG2020. The mBM also returned a probability distribution, but its origin was harder to understand due to the internal feature detection and weighting employed by neural networks. On the other hand, the lack of weighting of properties does have limitation. For example, if properties are added to the dataset that have a weak or non-existent relationship to other planet properties, then the proximity of these values within the parameter space will be the metaphorical "red herring" and could cause the kNN×KDE to select less informative neighbors for the distribution. Future developments for imputation methods could try to address this issue by considering adaptive metrics for the nearest-neighbor search algorithm. Learning an optimal metric would enable weighting the planet properties according to their relevance in the imputation of other properties, while still allowing for interpretation of the resulting distribution.

The information in the kNN×KDE probability distributions can be maximised by removing the neighbor cap that stops the kNN×KDE algorithm from considering the properties of more than 20 neighboring planets in the parameter space. As the kNN×KDE weights neighbors by proximity, more distant neighbors can be informative about possible but less probable values, which can be useful for understanding the expected demographics. As mentioned in section 4.8.1, averaging over the resultant distribution can result in a average bias. However, if the single point imputation is not required, or manually selected, this is not an issue.

4.8.3 A generative model

The final machine learning investigation was to use the kNN×KDE as a generative model to create a population of 10,000 synthetic planets with the eight properties of the extended dataset, and use this to identify clusters within the multidimensional parameter space. Of the six clusters identified, several groups were consistent with established planet classes. In particular, two of the largest clusters corresponded to hot Jupiters and super Earths. The remaining four identified groups were cooler gas giants on elliptical orbits, super Earths on short but elliptical orbits, and two classes of Neptune-sized planets, one around high metallicity stars with short orbit periods, and those on slightly longer orbits. Such clusters could indicate distinct evolutionary pathways, but the cluster classification is not always easy to determine and open for debate.

Chapter 5

Hierarchical Triple Systems Stability with CNNs

This chapter presents additional work done in collaboration with Alessandro Alberto Trani, postdoctoral researcher in astrophysics at the University of Tōkyō at the time. In this work, we developed a Convolutional Neural Network (CNN) to predict the stability of hierarchical triple systems, known for exhibiting a chaotic behaviour.

5.1 Context

The gravitational 3-body problem is a famous long-standing puzzle in astronomy and celestial mechanics, notable for having no closed-form solution to date. Besides its relevance to mathematical physics, the 3-body problem finds various applications in modern astrophysics, e.g. shaping the architecture of planetary systems [153].

In this work, we restrict our study to hierarchical triple systems, where an inner binary orbits another distant body (see Figure 5.1). Modeling the evolution of hierarchical triple systems involves the numerical integration of the Newtonian equations of motion, which can be computationally very expensive. Also, the chaotic nature of 3-body systems implies that accumulating integration errors inevitably lead to imprecision, which prevents from unequivocally predicting the fate of the system.

For these reasons, we decided to develop a CNN to predict the stability of hierarchical triple systems. The exercise boils down to a classification problem between "Stable" versus "Unstable" hierarchical systems, where the input data consists of several time series representing the evolution of the (combination of) Keplerian elements for the triple hierarchical system.

A review published in 2019 by Fawaz et al. showed that 1-dimensional CNNs are powerful tools providing state-of-the-art results for time series classification tasks [55]. Not only can the various CNN filters learn arbitrary time-invariant features, but also they allow to detect local discriminating characteristics useful for classification. We developed two CNN architecture types, and used various time series as input to assess the relative performances of our models.

Training data has been generated using TSUNAMI, a modern regularized code to evolve few-body self-gravitating systems developed by Alessandro Alberto Trani [154, 155].

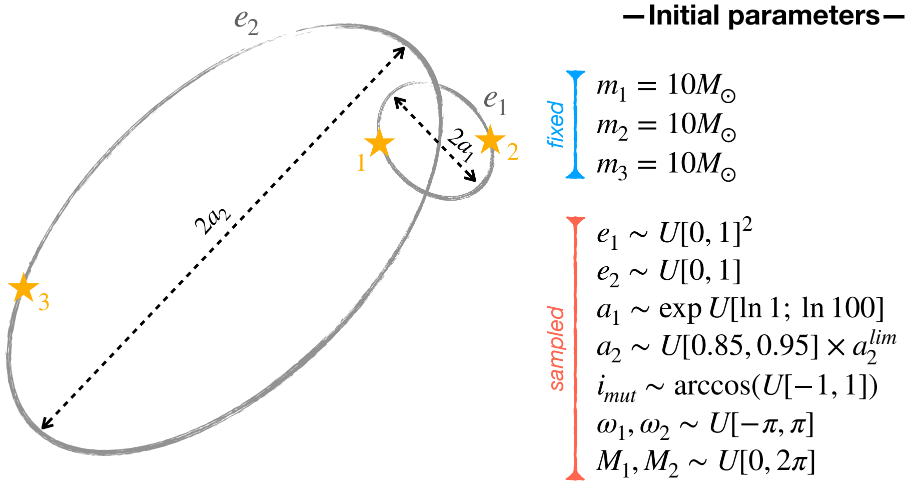

e_2
e_1
$2a_1$
$2a_2$
1
2
3
—Initial parameters—
fixed
$m_1 = 10M_\odot$
$m_2 = 10M_\odot$
$m_3 = 10M_\odot$
sampled
$e_1 \sim U[0,1]^2$
$e_2 \sim U[0,1]$
$a_1 \sim \exp U[\ln 1; \ln 100]$
$a_2 \sim U[0.85, 0.95] \times a_2^{lim}$
$i_{mut} \sim \arccos(U[-1,1])$
$\omega_1, \omega_2 \sim U[-\pi, \pi]$
$M_1, M_2 \sim U[0, 2\pi]$

Figure 5.1: Initial setup for the simulated hierarchical triple systems. The sampling range for the initial parameters are shown on the right. Body masses are fixed to $10\,M_\odot$, but the scale-free nature of Newtonian mechanics allows for arbitrary rescaling of the masses.

During the data generation process, we noticed extremely short-lived and completely stable systems to be over-represented although not particularly relevant for training. To create a better training dataset, we purposefully sampled outer semi-major axis near the Mardling and Aarseth stability criterion [156] in order to prevent from the simulation of obviously stable or unstable systems. In spite of this strategy, we faced imbalanced datasets problems especially towards short-lived systems, which called for caution during training [157].

During training, we tasked the CNNs to differentiate between “Stable” and “Unstable” systems, by providing only the first 0.5 % of the time series. If successful, using our CNN model allows to evolve the system for only 0.5 % of its final integration time, therefore predicting the stability of hierarchical triple systems 200 times faster.

5.2 Published article

Lalande Florian and Trani Alessandro Alberto. *Predicting the Stability of Hierarchical Triple Systems with Convolutional Neural Networks.* The Astrophysical Journal (ApJ) 938, 2022. ISSN 0004-637X.

5.3 Conclusion

Our best CNN model has an AUC (Area Under the ROC curve) of 0.956, making it robust for the classification of new unseen data [158]. We showed that the time-series evolution of the Keplerian elements allows to accurately predict the long term survival of hierarchical triple systems. Most notably, the inner and outer eccentricities provide relevant information for the stability prediction.

Besides, we used our simulated hierarchical triple systems to assess the stability prediction of Mushkin and Katz (2020) [159], which models the disruption of hierarchical triple systems via a random walk process. We found that the Mushkin and Katz stability criterion qualitatively follows the observed disruption times presented in our training dataset, but that quantitatively overestimates instability.

As the evolution of hierarchical triple systems remains chaotic, it is very challenging to predict. The great performances of our model could be further investigated, potentially revealing insight into the triggers towards disruption. The trained models and the Python scripts are available on GitLab[1].

[1] https://gitlab.com/aatrani/triple-stability-classifier

Chapter 6

Symbolic Regression with Transformer Models

This last chapter of my thesis covers a slightly different project. Between May and September 2023, I had the chance to do an internship at OMRON SINIC X (OSX), in Tōkyō, under the supervision of Yoshitaka Ushiku and Ryō Igarashi. My work consisted in developing Transformer models tailored for Symbolic Regression.

Although not exactly an astrophysics application, this work was surprisingly reminiscent to the Neptune project on one hand, and to my main PhD theme on numerical tabular data on the other hand. Moreover, the proposed Transformer models aim at automatically rediscover laws of physics taken from the Feynman Lectures on Physics Series, which include many astrophysics ones.

6.1 Context

Machine Learning models are often criticized for being black-box models, which implies that it is very complicated (if not impossible) to interpret their outputs. Symbolic Regression (SR) aims at solving this problem: it searches the space of mathematical expressions for an interpretable analytical formula that can explain a given dataset [85].

SR originated in the field of Genetic Programming [160], and state-of-the-art SR algorithms still use GP approaches. However, these approaches can be computationally expensive, and Machine Learning approaches have recently gained a lot of attention, despite poorer performances [161–163].

Until recently, there was no common benchmark for SR algorithms. In 2021, La Cava et al. proposed SRBench, a living and unified framework to test SR methods [164]. But last year, Matsubara et al. have pointed at several flaws of SRBench: inappropriate handling of constants, unrealistic sampling process, lack of diversity in orders of magnitude, the systematic treatment of integers as continuous variables, and the fact that only relevant variables are passed to the model [165]. To address these problems, Matsubara et al. proposed their own datasets: the SRSD datasets (Symbolic Regression for Scientific Discovery) based on the equations available in the Feynman Lectures on Physics Series [166]. The SRSD datasets include 120 datasets split into three groups with increasing difficulty: 30 easy, 40 medium, and 50 hard datasets. An example for an easy equation is the torque $\tau = rF\sin\theta$; for a medium equation is the 2-d Euclidean distance $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$; and for a hard equation is the Gaussian probability distribution $f = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{\theta^2}{2\sigma^2}\right)$.

Besides, Matsubara et al. also propose the normalized tree edit distance as a new evaluation metric for SR tasks, and defined as:

$$\tilde{d}(f_{\text{pred}}; f_{\text{true}}) = \min\left(1; \frac{d(f_{\text{pred}}; f_{\text{true}})}{|f_{\text{true}}|}\right)$$

where $d(f_{\text{pred}}; f_{\text{true}})$ is the tree edit distance computed with the Zhang and Shasha algorithm [167] between the predicted f_{pred} and the ground-truth f_{true} equations represented as trees, and $|f_{\text{true}}|$ is the number of tokens (or tree nodes) for the ground-truth equation.

My role during this internship was to develop a new Transformer model for Symbolic Regression, and assess its performances on the SRSD datasets using the newly proposed tree-edit distance. I was given complete freedom to build whatever architecture I pleased. Therefore, I decided to propose three Encoder architectures with increasing complexity, but at the cost of permutation equivariance with respect to the columns, a desirable property for numerical datasets.

6.2 Published article

Lalande Florian, Yoshitomo Matsubara, Naoya Chiba, Tatsunori Taniai, Ryo Igarashi, and Yoshitaka Ushiku. *A Transformer Model for Symbolic Regression towards Scientific Discovery*, accepted at NeurIPS 2023 AI for Science Workshop.

I discussed these results in an oral presentation at the NeurIPS 2023 AI for Science Workshop in New Orleans (Louisiana, US) on December 16, 2023.

6.3 Conclusion

The proposed architectures are presented in Figure 6.1. The difference between the three architectures lies in the Transformer encoder layers, and is further detailed in the appendix of the accepted manuscript [168].

Unlike traditional Transformer models, the input of the encoder does not consist in tokens anymore, but is a tabular dataset. Therefore, I propose three encoder architectures to work with numerical data, namely `MLP`, `Att`, and `Mix`. The innermost dimension of the model is d_{model}. The decoder works in an auto-regressive fashion and outputs probabilities for tokens in a vocabulary of size $v = 20$. Our best model uses the `Mix` architecture with $N_{\text{enc.}} = 4$, $N_{\text{dec.}} = 8$, and $d_{\text{model}} = 512$, for a total of about 9,620,000 trainable parameters.

We could show that our best model requires the highest flexibility level, although does not preserve the column permutation equivariance property of numerical datasets. A tentative explanation for this observation is that greater flexibility allows for better internal representations (in the latent space of the model). For good enough internal representations, we might even expect the column equivariance property to be automatically rediscovered, therefore unnecessary to enforce it. Further investigating the behaviour of

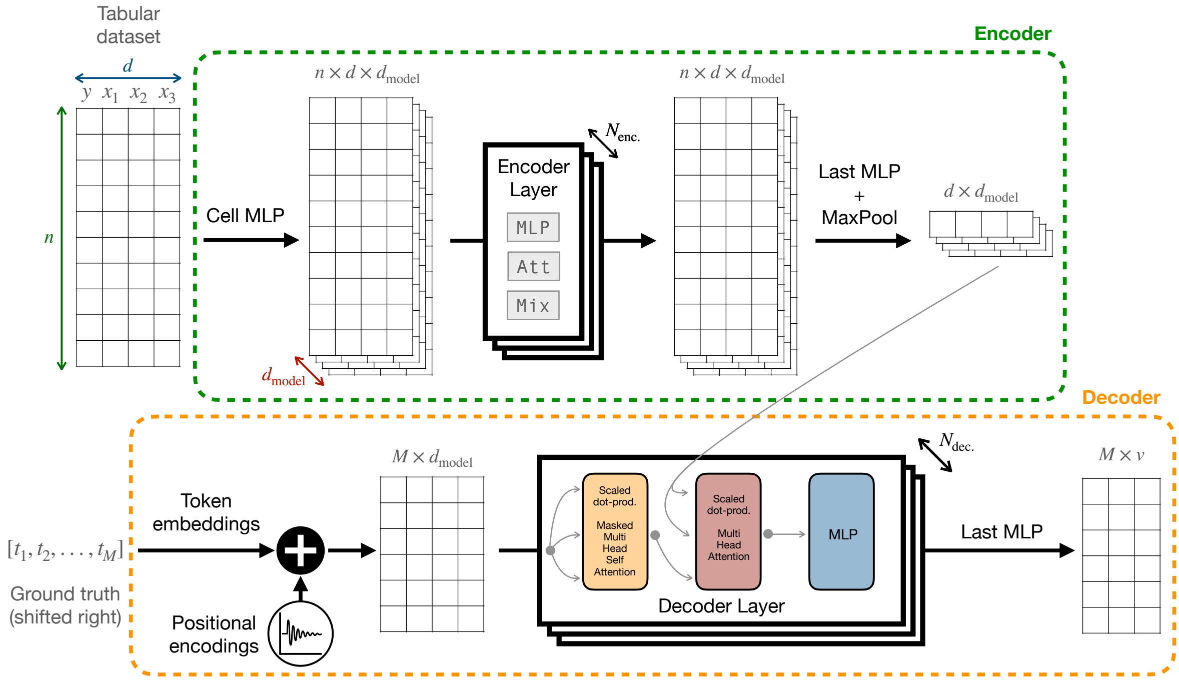

Tabular dataset
d
y x_1 x_2 x_3
n
Encoder
$n \times d \times d_{\text{model}}$
Cell MLP
d_{model}
Encoder Layer
MLP
Att
Mix
$N_{\text{enc.}}$
$n \times d \times d_{\text{model}}$
Last MLP + MaxPool
$d \times d_{\text{model}}$
Decoder
$[t_1, t_2, \ldots, t_M]$
Ground truth (shifted right)
Token embeddings
Positional encodings
$M \times d_{\text{model}}$
Scaled dot-prod. Masked Multi Head Self Attention
Scaled dot-prod. Multi Head Attention
MLP
Decoder Layer
$N_{\text{dec.}}$
Last MLP
$M \times v$

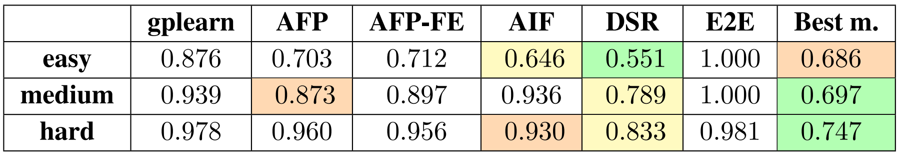
Figure 6.1: Architecture of the proposed Transformer models for Symbolic Regression. I propose three encoder architectures: `MLP`, `Att`, or `Mix`. The decoder is a standard Transformer decoder and is the same in all cases. During training, the encoder receives the tabular dataset and the decoder receives the ground-truth sequence of tokens, used with teacher-forcing method. During inference, the decoder is on its own and predicts tokens in an auto-regressive manner.

the proposed encoder architectures and exploring the patterns learned in the latent space constitute potentially interesting future work for SR with Transformer models.

Once pre-trained using synthetic generated datasets in a supervised learning fashion, we tested our best Transformer model against other traditional SR algorithms: four GP-based approaches (gplearn, AFP, AFP-FE, and AIF), one RNN-based risk-seeking network (DSR), and another recent Transformer model for Symbolic Regression (E2E). Additional details and references are given in our manuscript [168].

Figure 6.2 compares the performances of the seven SR algorithms using the SRSD datasets and computing the error via the normalized tree edit distance for evaluation metric, as suggested by Matsubara et al. [165]. On top of providing almost instantaneous inference, we showed that our best Transformer model yields very good results. Note that DSR (for Deep-Symbolic Regression), in spite of being a neural-network based approach, has to be trained from scratch for every new numerical dataset. Only E2E and our proposed Transformer model are "fully pre-trained" ANN models which can deliver instantaneous estimates.

In conclusion, SR remains a very complicated problem, mostly because of the vast searching space for mathematical expressions. Its application on real datasets constitute an even more challenging problem, as data might be missing, noisy, or censored. Throughout this work, I realized that basic assumptions end up playing a crucial role,



	gplearn	AFP	AFP-FE	AIF	DSR	E2E	Best m.
easy	0.876	0.703	0.712	0.646	0.551	1.000	0.686
medium	0.939	0.873	0.897	0.936	0.789	1.000	0.697
hard	0.978	0.960	0.956	0.930	0.833	0.981	0.747

Figure 6.2: Aggregated performances on the SRSD datasets. Our best Transformer model (with the `Mix` encoder) outperforms other traditional SR methods on the medium and hard SRSD datasets, and provides competitive performances on the easy SRSD datasets.

e.g. the sampling range, the treatment of variables, the chosen dictionary of tokens, the representation of the ground-truth, or the evaluation procedure. While pre-trained Transformer models have a serious computational advantage over GP algorithms, they remain less flexible and we hope that this work can pave the way towards more powerful and flexible Transformer models for Symbolic Regression. Our code has been made available on GitHub and includes a user-friendly Jupyter notebook to play with our models[1].

[1] `https://github.com/omron-sinicx/transformer4sr`

Conclusion & Future Directions

Summary of Contributions and Insights

Data imputation algorithms are as useful tools as missing data is a pervasive problem. This PhD thesis proposes to rethink data imputation for numerical datasets, and uses the NASA Exoplanet Archive as motivating application. The major contributions of this work are three-fold.

(i) Reaffirm the superiority of statistical methods over Deep-Learning for numerical data imputation. This work proposes a comprehensive overview of traditional and more recent numerical data imputation methods, in various missing data scenarios and with various missing rates. The results presented in this thesis are in agreement with the emerging scientific consensus: Artificial Neural Networks (ANNs) are not mature enough for tabular datasets. Besides, traditional statistical methods offer easier interpretation and are often less computationally expensive.

(ii) Propose a novel numerical data imputation algorithm. Choosing a point estimate always implies loosing a lot of information. Instead of returning a point estimate, I proposed the kNN×KDE, a numerical data imputation algorithm that returns a multivariate probability distribution for each observation, given the observed values. This allows for more complex subsequent analysis, flexible ways to perform the final imputation, or the possibility to sample from that distribution.

(iii) Understand the high-dimensional distribution of planets and accurately estimate missing properties. I applied the kNN×KDE to the NASA Exoplanet Archive to estimate the missing masses and radii of planets. The analysis of the returned distributions shed light on the multi-dimensional demographics of exoplanets, existing and suspected trends were re-discovered (e.g. relations between star metallicitiy, orbital eccentricity, number of planets, and mass), and planet groups have been automatically re-identified.

Besides my main theme on numerical data imputation, this PhD thesis provides additional insights regarding the use of Machine Learning for Astrophysics. In particular, Machine Learning has the potential to leverage complex data structures (i.e. not numerical tabular datasets), and provide with tangible scientific applications. Namely, I showed how Convolutional Neural Networks (CNNs) can analyze time series to predict the long-term stability of hierarchical triple systems, known for being chaotic. In addition, and when used correctly, Machine Learning models can also save tremendous computational time. For instance, Transformer models can yield nearly instantaneous predictions for Symbolic Regression tasks, while traditional Genetic Programming approaches may take few hours for each dataset. Similarly, the CNNs developed for hierarchical triple systems stability prediction can spare from computationally expensive numerical integrations.

Future Work and Recommendations

In spite of the "Universal Approximation Theorem" – which guarantee that any continuous function can be arbitrarily well approximated over a finite domain by appropriate ANNs with enough number of parameters – there is growing evidence that ANNs are not mature enough to tackle numerical datasets. For tabular data imputation tasks, Generative Adversarial Networks (GANs) appear like a promising avenue, but do not yet provide satisfactory enough results, because of structural limitations of ANNs included by design. As statistical methods remain the gold standard, new research and development should be done in the area of ANNs for numerical datasets. I believe that ANNs will eventually supplant traditional methods for tabular datasets, like it already happened with images, text, video, or time series data (probably because these data type are much more sparse). But this will likely require disruptive innovations to allow ANNs architectures for better processing of tabular numerical data.

Meanwhile, data imputation of numerical datasets remains a ubiquitous problem for data practitioners. Instead of estimating each missing value with a point estimate, I recommend going one step further and aim for the density estimation of missing values, which allows to capture the complexity of distributions (several modes, variance, skewness) as well as potentially complex multi-dimensional relations, invisible on univariate distributions. In this respect, **I propose and recommend the use of the kNN×KDE**, inspired by the kNN algorithm with kernel methods. I showed that the kNN×KDE is on par with current state-of-the-art imputation methods when providing point estimates while additionally offering multi-dimensional density estimates. It is therefore at least as good as other traditional numerical data imputation algorithms.

That said, there is still a lot of room for improvement regarding numerical data imputation methods. Firstly, I decided to build upon the standard kNN-Imputer as it offers a simple framework, but I am convinced that using Random Forests as the basis for subsequent kernel density estimations might lead to even better results (but maybe more complicated to implement). Next, the proposed kNN×KDE can be further adjusted to work with specific use cases: this has been shown when applied to the NASA Exoplanet Archive, e.g. with the addition of new predictors whose imputation does not matter. I tried to leave the original framework of the kNN×KDE as flexible as possible for specific use cases, which may lead to the development of much better methods that this one. Finally, and maybe of highest importance, the paradigm of numerical data imputation has to be reevaluated. Similar to other ML domains, evaluation metrics have to be closer to human judgments: the BLEU Score in NLP, Inception Score for Computer Vision, or Reward/Punishment Scores in Reinforcement Learning. Under those circumstances, why using the RMSE as only evaluation metric for numerical data imputation? This seems too restrictive, and I propose the log-likelihood instead, although not flawless either. The development of new metrics to assess the quality of imputed values in numerical dataset seems to be an area to prioritize future research.

Bibliography

[1] Michel Mayor and Didier Queloz. A jupiter-mass companion to a solar-type star. *Nature*, 378, 1995. ISSN 00280836. doi: 10.1038/378355a0.

[2] Tod R. Lauer et al. New horizons observations of the cosmic optical background. *The Astrophysical Journal*, 906, 2021. ISSN 0004-637X. doi: 10.3847/1538-4357/abc881.

[3] Guillem Anglada-Escudé et al. A terrestrial planet candidate in a temperate orbit around proxima centauri. *Nature*, 536, 2016. ISSN 14764687. doi: 10.1038/nature19106.

[4] J. P. Faria et al. A candidate short-period sub-earth orbiting proxima centauri. *Astronomy and Astrophysics*, 658, 2022. ISSN 14320746. doi: 10.1051/0004-6361/202142337.

[5] Mario Damasso et al. A low-mass planet candidate orbiting proxima centauri at a distance of 1.5 au. *Science Advances*, 6, 2020. ISSN 23752548. doi: 10.1126/sciadv.aax7467.

[6] Christopher J. Shallue and Andrew Vanderburg. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, 155, 2018. ISSN 0004-6256. doi: 10.3847/1538-3881/aa9e09.

[7] Yan Liang, Jakob Robnik, and Uroš Seljak. Kepler-90: Giant transit-timing variations reveal a super-puff. *The Astronomical Journal*, 161, 2021. ISSN 0004-6256. doi: 10.3847/1538-3881/abe6a7.

[8] David Kipping. An independent analysis of the six recently claimed exomoon candidates. *The Astrophysical Journal Letters*, pages 1–16, 2020. URL `http://arxiv.org/abs/2008.03613`.

[9] Mary Anne Limbach, Johanna M. Vos, Joshua N. Winn, René Heller, Jeffrey C. Mason, Adam C. Schneider, and Fei Dai. On the detection of exomoons transiting isolated planetary-mass objects. *The Astrophysical Journal Letters*, 918, 2021. ISSN 2041-8205. doi: 10.3847/2041-8213/ac1e2d.

[10] David Kipping, Steve Bryson, Chris Burke, Jessie Christiansen, Kevin Hardegree-Ullman, Billy Quarles, Brad Hansen, Judit Szulágyi, and Alex Teachey. An exomoon survey of 70 cool giant exoplanets and the new candidate kepler-1708 b-i. *Nature Astronomy*, 6, 2022. ISSN 23973366. doi: 10.1038/s41550-021-01539-1.

[11] M. Woolfson. The origin and evolution of the solar system. *Astronomy and Geophysics*, 41, 2000. ISSN 13668781. doi: 10.1046/j.1468-4004.2000.00012.x.

[12] Elizabeth Tasker. *The Planet Factory*. Bloomsbury, 2017. doi: 10.5040/9781472956446.

[13] Michael Perryman. *The Exoplanet Handbook, Second Edition*. Cambridge University Press, 2018. doi: 10.1017/9781108304160.

[14] James F. Kasting, Daniel P. Whitmire, and Ray T. Reynolds. Habitable zones around main sequence stars. *Icarus*, 101, 1993. ISSN 10902643. doi: 10.1006/icar.1993.1010.

[15] Ravi Kumar Kopparapu et al. Habitable zones around main-sequence stars: New estimates. *Astrophysical Journal*, 765, 2013. ISSN 15384357. doi: 10.1088/0004-637X/765/2/131.

[16] Elizabeth Tasker et al. The language of exoplanet ranking metrics needs to change, 2017. ISSN 23973366.

[17] Erik A. Petigura, Andrew W. Howard, and Geoffrey W. Marcy. Prevalence of earth-size planets orbiting sun-like stars. *Proceedings of the National Academy of Sciences of the United States of America*, 110, 2013. ISSN 00278424. doi: 10.1073/pnas.1319909110.

[18] Webb S. *If the Universe is Teeming with Aliens... Where is Everybody?* Copernicus New York, springer edition, 2004. doi: 10.1007/b97464.

[19] GMHJ Habets and J R W Heintze. Empirical bolometric corrections for the main-sequence. *Astronomy and Astrophysics Supplement Series*, 46, 1981.

[20] M. Asplund. The new solar abundances - part i: the observations. *Communications in Asteroseismology*, 147, 2007. ISSN 1021-2043. doi: 10.1553/cia147s76.

[21] Event Horizon Telescope Collaboration. First sagittarius a* event horizon telescope results. i. the shadow of the supermassive black hole in the center of the milky way. *The Astrophysical Journal Letters*, 930, 2022. ISSN 2041-8205. doi: 10.3847/2041-8213/ac6674.

[22] François Fressin et al. The false positive rate of kepler and the occurrence of planets. *Astrophysical Journal*, 766, 2013. ISSN 15384357. doi: 10.1088/0004-637X/766/2/81.

[23] Jerome A. Orosz et al. Discovery of a third transiting planet in the kepler-47 circumbinary system. *The Astronomical Journal*, 157, 2019. ISSN 00046256. doi: 10.3847/1538-3881/ab0ca0.

[24] Elisa V. Quintana and Jack J. Lissauer. Terrestrial planet formation surrounding close binary stars. *Icarus*, 185, 2006. ISSN 00191035. doi: 10.1016/j.icarus.2006.06.016.

[25] Tristan Guillot, David J Stevenson, William B Hubbard, and Didier Saumon. *The Interior of Jupiter*, chapter 3. Cambridge University Press, 2004. ISBN 978-0-521-81808-7.

[26] Geoffrey Marcy, R. Paul Butler, Debra Fischer, Steven Vogt, Jason T. Wright, Chris G. Tinney, and Hugh R.A. Jones. Observed properties of exoplanets: Masses, orbits, and metallicities. *Progress of Theoretical Physics Supplement*, 158, 2005. ISSN 03759687. doi: 10.1143/PTPS.158.24.

[27] David Charbonneau, Timothy M. Brown, David W. Latham, and Michel Mayor. Detection of planetary transits across a sun-like star. *The Astrophysical Journal*, 529, 2000. ISSN 0004637X. doi: 10.1086/312457.

[28] M. Auvergne et al. The corot satellite in flight: Description and performance. *Astronomy and Astrophysics*, 506, 2009. ISSN 14320746. doi: 10.1051/0004-6361/200810860.

[29] William J. Borucki et al. Kepler planet-detection mission: Introduction and first results. *Science*, 327, 2010. ISSN 00368075. doi: 10.1126/science.1185402.

[30] George R. Ricker et al. Transiting exoplanet survey satellite. *Journal of Astronomical Telescopes, Instruments, and Systems*, 1, 2014. ISSN 2329-4124. doi: 10.1117/1.jatis.1.1.014003.

[31] Otto Struve. Proposal for a project of high-precision stellar radial velocity work. *The Observatory*, 72:199–200, 10 1952.

[32] M. Mayor et al. Setting new standards with harps. *The Messenger (ISSN0722-6691)*, 2003.

[33] F. Pepe et al. Espresso: The next european exoplanet hunter. *Astronomische Nachrichten*, 335, 2014. ISSN 00046337. doi: 10.1002/asna.201312004.

[34] J. P. Beaulieu et al. Discovery of a cool planet of 5.5 earth masses through gravitational microlensing. *Nature*, 439, 2006. ISSN 14764687. doi: 10.1038/nature04441.

[35] Christian Marois, Bruce Macintosh, Travis Barman, B. Zuckerman, Inseok Song, Jennifer Patience, David Lafrenière, and René Doyon. Direct imaging of multiple planets orbiting the star hr 8799. *Science*, 322, 2008. ISSN 00368075. doi: 10.1126/science.1166585.

[36] David M. Kipping and Emily Sandford. Observational biases for transiting planets. *Monthly Notices of the Royal Astronomical Society*, 463, 2016. ISSN 13652966. doi: 10.1093/mnras/stw1926.

[37] Jessie L. Christiansen, Bruce D. Clarke, Christopher J. Burke, Shawn Seader, Jon M. Jenkins, Joseph D. Twicken, Joseph D. Catanzarite, Jeffrey C. Smith, Natalie M. Batalha, Michael R. Haas, Susan E. Thompson, Jennifer R. Campbell, Anima Sabale, and A. K.M.Kamal Uddin. Measuring transit signal recovery in the kepler pipeline. ii. detection efficiency as calculated in one year of data. *Astrophysical Journal*, 810, 2015. ISSN 15384357. doi: 10.1088/0004-637X/810/2/95.

[38] Yiannis Tsapras. Microlensing searches for exoplanets. *Geosciences (Switzerland)*, 8, 2018. ISSN 20763263. doi: 10.3390/geosciences8100365.

[39] X. Dumusque, N. C. Santos, S. Udry, C. Lovis, and X. Bonfils. Planetary detection limits taking into account stellar noise: Ii. effect of stellar spot groups on radial-velocities. *Astronomy and Astrophysics*, 527, 2011. ISSN 00046361. doi: 10.1051/0004-6361/201015877.

[40] François Bouchy and Fabien Carrier. Present observational status of solar-type stars. *Astrophysics and Space Science*, 284, 2003. ISSN 0004640X. doi: 10.1023/A:1023216124310.

[41] T. R. Bedding. Observations of solar-like oscillations. *Communications in Asteroseismology*, 150:106–114, 2007. ISSN 1021-2043. doi: 10.1553/cia150s106. URL `http://dx.doi.org/10.1553/cia150s106`.

[42] D. Del Moro. Solar granulation properties derived from three different time series. *Astronomy and Astrophysics*, 428, 2004. ISSN 00046361. doi: 10.1051/0004-6361:20040466.

[43] D. Del Moro, F. Berrilli, T. L. Duvall, and A. G. Kosovichev. Dynamics and structure of supergranulation. *Solar Physics*, 221, 2004. ISSN 00380938. doi: 10.1023/B:SOLA.0000033363.15641.8f.

[44] N. Meunier, M. Desort, and A. M. Lagrange. Using the sun to estimate earth-like planets detection capabilities i. impact of cold spots. *Astronomy and Astrophysics*, 512, 2010. ISSN 14320746. doi: 10.1051/0004-6361/200913551.

[45] Carolus J. Schrijver. Simulations of the photospheric magnetic activity and outer atmospheric radiative losses of cool stars based on characteristics of the solar magnetic field. *The Astrophysical Journal*, 547, 2001. ISSN 0004-637X. doi: 10.1086/318333.

[46] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5, 1943. ISSN 00074985. doi: 10.1007/BF02478259.

[47] Donald Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press, first edition, 1949. ISBN 978-0805843002.

[48] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 1958. ISSN 0033295X. doi: 10.1037/h0042519.

[49] Marvin Lee Minsky and Seymour Aubrey Papert. *Perceptrons: an introduction to computational geometry*. MIT Press, second edition, 1969. ISBN 978-0262534772.

[50] Paul John Werbos. *The Roots of Backpropagation*. Wiley-Interscience, first edition, 1994. ISBN 978-0471598978.

[51] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1, 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541.

[52] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2, 1989. ISSN 09324194. doi: 10.1007/BF02551274.

[53] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei Fei Li. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014. doi: 10.1109/CVPR.2014.223.

[54] A. Peel, F. Lalande, J.-L. Starck, V. Pettorino, J. Merten, C. Giocoli, M. Meneghetti, and M. Baldi. Distinguishing standard and modified gravity cosmologies with machine learning. *arXiv*, 2018.

[55] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33, 2019. ISSN 1573756X. doi: 10.1007/s10618-019-00619-1.

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9, 1997. ISSN 08997667. doi: 10.1162/neco.1997.9.8.1735.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

[58] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

[59] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 28492–28518. PMLR, 23–29 Jul 2023.

[60] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 3, 2014.

[61] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. doi: 10.23915/distill.00033. https://distill.pub/2021/gnn-intro.

[62] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks, 2011.

[63] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11, 2010. ISSN 15324435.

[64] Richard Bellman. *Dynamic Programming.* Dover Publications, reprint edition, 1956. ISBN 978-0486428093.

[65] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[66] Francis Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15, 1886. ISSN 09595295. doi: 10.2307/2841583.

[67] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1, 1986. ISSN 15730565. doi: 10.1023/A:1022643204877.

[68] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001. ISSN 08856125. doi: 10.1023/A:1010933404324.

[69] Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57, 1989. ISSN 03067734. doi: 10.2307/1403797.

[70] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 1901. ISSN 1941-5982. doi: 10.1080/14786440109462720.

[71] Hanyu Xiang, Qin Zou, Muhammad Ali Nawaz, Xianfeng Huang, Fan Zhang, and Hongkai Yu. Deep learning for image inpainting: A survey. *Pattern Recognition*, 134, 2023. ISSN 00313203. doi: 10.1016/j.patcog.2022.109046.

[72] Roderick J.A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data.* Wiley, 2014. doi: 10.1002/9781119013563.

[73] Philip L. Roth. Missing data: a conceptual review for applied psychologists. *Personnel Psychology*, 47, 1994. ISSN 17446570. doi: 10.1111/j.1744-6570.1994.tb01736.x.

[74] Frank J. Molnar, Brian Hutton, and Dean Fergusson. Does analysis using "last observation carried forward" introduce bias in dementia research?, 2008. ISSN 14882329.

[75] Graham Kalton and Daniel Kasprzyk. Imputing for missing survey responses, 1982.

[76] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17, 2001. ISSN 13674803. doi: 10.1093/bioinformatics/17.6.520.

[77] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45, 2011. ISSN 15487660. doi: 10.18637/jss.v045.i03.

[78] Daniel J. Stekhoven and Peter Bühlmann. Missforest-non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28, 2012. ISSN 13674803. doi: 10.1093/bioinformatics/btr597.

[79] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Gain: Missing data imputation using generative adversarial nets. *35th International Conference on Machine Learning, ICML 2018*, 13:9042–9051, 2018.

[80] Alfredo Nazábal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107, 2020. ISSN 00313203. doi: 10.1016/j.patcog.2020.107501.

[81] Sebastian Jäger, Arndt Allhorn, and Felix Bießmann. A benchmark for data imputation methods. *Frontiers in Big Data*, 4, 2021. ISSN 2624909X. doi: 10.3389/fdata.2021.693674.

[82] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on tabular data? In *Advances in Neural Information Processing Systems*, 2022.

[83] Isaac Newton. *Philosophiæ Naturalis Principia Mathematica.* Halley, England, 1687.

[84] A. Danjon. Le centenaire de la decouverte de neptune. *Ciel et Terre*, 62:369, January 1946.

[85] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324, 2009. ISSN 00368075. doi: 10.1126/science.1165893.

[86] Steven Cheng Xian Li, Benjamin M. Marlin, and Bo Jiang. Misgan: Learning from incomplete data with generative adversarial networks. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[87] Chao Ma and Cheng Zhang. Identifiable generative models for missing not at random data imputation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=bGXIX-CVzrq`.

[88] Gustavo E. A. P. A. Batista and Maria C. Monard. A study of k-nearest neighbour as an imputation method. *Frontiers in Artificial Intelligence and Applications*, 87, 2002.

[89] Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. From predictive methods to missing data imputation: An optimization approach. *Journal of Machine Learning Research*, 18, 2018. ISSN 15337928.

[90] Jason Poulos and Rafael Valle. Missing data imputation for supervised learning. *Applied Artificial Intelligence*, 32, 2018. ISSN 10876545. doi: 10.1080/08839514.2018.1448143.

[91] Anil Jadhav, Dhanya Pramod, and Krishnan Ramanathan. Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, 33, 2019. ISSN 10876545. doi: 10.1080/08839514.2019.1637138.

[92] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions. *OpenAccess*, 2020. doi: 10.48550/arXiv.2005.00065.

[93] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4413–4423. PMLR, 09–15 Jun 2019.

[94] Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. not-{miwae}: Deep generative modelling with missing not at random data. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=tu29GQT0JFy`.

[95] Florian Lalande and Kenji Doya. Numerical data imputation: Choose knn over deep learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 13590 LNCS, 2022. doi: 10.1007/978-3-031-17849-8_1.

[96] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27, 1956. ISSN 0003-4851. doi: 10.1214/aoms/1177728190.

[97] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33, 1962. ISSN 0003-4851. doi: 10.1214/aoms/1177704472.

[98] D. M. Titterington and G. M. Mill. Kernel-based density estimates from incomplete data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45, 1983. doi: 10.1111/j.2517-6161.1983.tb01249.x.

[99] Richard Leibrandt and Stephan Günnemann. Making kernel density estimation robust towards missing values in highly incomplete multivariate data without imputation. In *SIAM International Conference on Data Mining, SDM 2018*, 2018. doi: 10.1137/1.9781611975321.84.

[100] John K. Dixon. Pattern recognition with partly missing data. *IEEE Transactions on Systems, Man and Cybernetics*, 9, 1979. ISSN 21682909. doi: 10.1109/TSMC.1979.4310090.

[101] Lalande Florian and Doya Kenji. Numerical data imputation for multimodal data sets: A probabilistic nearest-neighbor kernel density approach. *Transactions on*

Machine Learning Research, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=KqR3rgooXb. Reproducibility Certification.

[102] Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *Journal of Machine Learning Research*, 16, 2015. ISSN 15337928.

[103] Debra A. Fischer and Jeff Valenti. The Planet-Metallicity Correlation. *APJ*, 622 (2):1102–1117, April 2005. doi: 10.1086/428383.

[104] Courtney D. Dressing and David Charbonneau. The Occurrence Rate of Small Planets around Small Stars. *APJ*, 767(1):95, April 2013. doi: 10.1088/0004-637X/767/1/95.

[105] Ares Osborn and Daniel Bayliss. Investigating the planet-metallicity correlation for hot Jupiters. *MNRAS*, 491(3):4481–4487, January 2020. doi: 10.1093/mnras/stz3207.

[106] Vardan Adibekyan et al. A compositional link between rocky exoplanets and their host stars. *Science*, 374(6565):330–332, October 2021. doi: 10.1126/science.abg8794.

[107] Edward M. Bryant, Daniel Bayliss, and Vincent Van Eylen. The occurrence rate of giant planets orbiting low-mass stars with TESS. *MNARS*, 521(3):3663–3681, May 2023. doi: 10.1093/mnras/stad626.

[108] M. Oshagh et al. Understanding stellar activity-induced radial velocity jitter using simultaneous K2 photometry and HARPS RV measurements. *AAP*, 606:A107, October 2017. doi: 10.1051/0004-6361/201731139.

[109] A. L. Wallace and M. J. Ireland. The likelihood of detecting young giant planets with high-contrast imaging and interferometry. *MNRAS*, 490(1):502–512, November 2019. doi: 10.1093/mnras/stz2600.

[110] Jingjing Chen and David Kipping. Probabilistic Forecasting of the Masses and Radii of Other Worlds. *APJ*, 834(1):17, January 2017. doi: 10.3847/1538-4357/834/1/17.

[111] Mary Anne Limbach and Edwin L. Turner. Exoplanet orbital eccentricity: Multiplicity relation and the Solar System. *Proceedings of the National Academy of Science*, 112(1):20–24, January 2015. doi: 10.1073/pnas.1406545111.

[112] Leslie A. Rogers. Most 1.6 Earth-radius Planets are Not Rocky. *APJ*, 801(1):41, March 2015. doi: 10.1088/0004-637X/801/1/41.

[113] Lauren M. Weiss and Geoffrey W. Marcy. The Mass-Radius Relation for 65 Exoplanets Smaller than 4 Earth Radii. *APJL*, 783(1):L6, March 2014. doi: 10.1088/2041-8205/783/1/L6.

[114] John Asher Johnson, Kimberly M. Aller, Andrew W. Howard, and Justin R. Crepp. Giant Planet Occurrence in the Stellar Mass-Metallicity Plane. *PASP*, 122(894): 905, August 2010. doi: 10.1086/655775.

[115] Elizabeth J. Tasker, Matthieu Laneuville, and Nicholas Guttenberg. Estimating planetary mass with deep learning. *The Astronomical Journal*, 159:41, 2020. ISSN 1538-3881. doi: 10.3847/1538-3881/ab5b9e.

[116] Aldo S. Bonomo et al. A giant impact as the likely origin of different twins in the Kepler-107 exoplanet system. *Nature Astronomy*, 3:416–423, February 2019. doi: 10.1038/s41550-018-0684-9.

[117] Cayman T. Unterborn, Steven J. Desch, Natalie R. Hinkel, and Alejandro Lorenzo. Inward migration of the TRAPPIST-1 planets as inferred from their water-rich compositions. *Nature Astronomy*, 2:297–302, March 2018. doi: 10.1038/s41550-018-0411-6.

[118] Jade C. Bond, David P. O'Brien, and Dante S. Lauretta. The Compositional Diversity of Extrasolar Terrestrial Planets. I. In Situ Simulations. *APJ*, 715(2):1050–1070, June 2010. doi: 10.1088/0004-637X/715/2/1050.

[119] Guttenberg N. Tasker E.J., Laneuville M. Data for 550 exoplanets using a neural network, 2020.

[120] Lauren M. Weiss et al. The California-Kepler Survey. V. Peas in a Pod: Planets in a Kepler Multi-planet System Are Similar in Size and Regularly Spaced. *The Astronomical Journal*, 155(1):48, January 2018. doi: 10.3847/1538-3881/aa9ff6.

[121] Sean N. Raymond. The Search for Other Earths: Limits on the Giant Planet Orbits That Allow Habitable Terrestrial Planets to Form. *APJL*, 643(2):L131–L134, June 2006. doi: 10.1086/505596.

[122] Sean N. Raymond, Thomas Quinn, and Jonathan I. Lunine. The formation and habitability of terrestrial planets in the presence of close-in giant planets. *Icarus*, 177(1):256–263, September 2005. doi: 10.1016/j.icarus.2005.03.008.

[123] Philip J. Armitage. A Reduced Efficiency of Terrestrial Planet Formation following Giant Planet Migration. *APJL*, 582(1):L47–L50, January 2003. doi: 10.1086/346198.

[124] M. Mousavi-Sadr, D. M. Jassur, and G. Gozaliasl. Revisiting mass-radius relationships for exoplanet populations: a machine learning insight. *MNRAS*, 525(3): 3469–3485, November 2023. doi: 10.1093/mnras/stad2506.

[125] Tuhin Ghosh and Sourav Chatterjee. Orbital architectures of Kepler multis from dynamical instabilities. *MNRAS*, 527(1):79–92, January 2024. doi: 10.1093/mnras/stad2962.

[126] M. Juric and S. Tremaine. Dynamical Relaxation by Planet-Planet Interactions as the Origin of Exoplanet Eccentricity Distribution. In D. Fischer, F. A. Rasio, S. E. Thorsett, and A. Wolszczan, editors, *Extreme Solar Systems*, volume 398 of *Astronomical Society of the Pacific Conference Series*, page 295, January 2008.

[127] Frederic A. Rasio and Eric B. Ford. Dynamical instabilities and the formation of extrasolar planetary systems. *Science*, 274:954–956, November 1996. doi: 10.1126/science.274.5289.954.

[128] Lars A. Buchhave, Bertram Bitsch, Anders Johansen, David W. Latham, Martin Bizzarro, Allyson Bieryla, and David M. Kipping. Jupiter Analogs Orbit Stars with an Average Metallicity Close to That of the Sun. *APJ*, 856(1):37, March 2018. doi: 10.3847/1538-4357/aaafca.

[129] Rebekah I. Dawson and Ruth A. Murray-Clay. Giant Planets Orbiting Metal-rich Stars Show Signatures of Planet-Planet Interactions. *APJL*, 767(2):L24, April 2013. doi: 10.1088/2041-8205/767/2/L24.

[130] Karl Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London Series I*, 58:240–242, January 1895.

[131] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans): Challenges, solutions, and future directions. *ACM Computing Surveys*, 54(3), 2021. ISSN 0360-0300. doi: 10.1145/3446374. URL `https://doi.org/10.1145/3446374`.

[132] J. D. Hartman et al. HAT-P-57b: A Short-period Giant Planet Transiting a Bright Rapidly Rotating A8V Star Confirmed Via Doppler Tomography. *The Astronomical Journal*, 150(6):197, December 2015. doi: 10.1088/0004-6256/150/6/197.

[133] Keivan G. Stassun, Karen A. Collins, and B. Scott Gaudi. Accurate Empirical Radii and Masses of Planets and Their Host Stars with Gaia Parallaxes. *The Astronomical Journal*, 153(3):136, March 2017. doi: 10.3847/1538-3881/aa5df3.

[134] J. T. Wright, G. W. Marcy, A. W. Howard, John Asher Johnson, T. D. Morton, and D. A. Fischer. The Frequency of Hot Jupiters Orbiting nearby Solar-type Stars. *The Astrophysical Journal*, 753(2):160, July 2012. doi: 10.1088/0004-637X/753/2/160.

[135] L. Borsato et al. HARPS-N radial velocities confirm the low densities of the Kepler-9 planets. *MNARS*, 484(3):3233–3243, April 2019. doi: 10.1093/mnras/stz181.

[136] Roberto Sanchis-Ojeda et al. Alignment of the stellar spin with the orbits of a three-planet system. *Nature*, 487(7408):449–453, July 2012. doi: 10.1038/nature11301.

[137] A. Mortier et al. K2-111: an old system with two planets in near-resonance. *MNARS*, 499(4):5004–5021, December 2020. doi: 10.1093/mnras/staa3144.

[138] Daisuke Kawata et al. JASMINE: Near-Infrared Astrometry and Time Series Photometry Science. *arXiv e-prints*, art. arXiv:2307.05666, July 2023. doi: 10.48550/arXiv.2307.05666.

[139] Steve B. Howell et al. Kepler-21b: A 1.6 R $_{Earth}$ Planet Transiting the Bright Oscillating F Subgiant Star HD 179070. *APJ*, 746(2):123, February 2012. doi: 10.1088/0004-637X/746/2/123.

[140] Michaël Gillon et al. Temperate Earth-sized planets transiting a nearby ultracool dwarf star. *Nature*, 533(7602):221–224, May 2016. doi: 10.1038/nature17448.

[141] Fabo Feng et al. 3D Selection of 167 Substellar Companions to Nearby Stars. *APJS*, 262(1):21, September 2022. doi: 10.3847/1538-4365/ac7e57.

[142] B. Enoch, A. Collier Cameron, and K. Horne. Factors affecting the radii of close-in transiting exoplanets. *AAP*, 540:A99, April 2012. doi: 10.1051/0004-6361/201117317.

[143] Geoffrey W. Marcy et al. Masses, Radii, and Orbits of Small Kepler Planets: The Transition from Gaseous to Rocky Planets. *APJS*, 210(2):20, February 2014. doi: 10.1088/0067-0049/210/2/20.

[144] Nikolay Nikolov et al. Ground-based Transmission Spectroscopy with VLT FORS2: Evidence for Faculae and Clouds in the Optical Spectrum of the Warm Saturn WASP-110b. *AJ*, 162(3):88, September 2021. doi: 10.3847/1538-3881/ac01da.

[145] Veselin B. Kostov et al. TIC 172900988: A Transiting Circumbinary Planet Detected in One Sector of TESS Data. *AJ*, 162(6):234, December 2021. doi: 10.3847/1538-3881/ac223a.

[146] R. Paul Butler, Geoffrey W. Marcy, Steven S. Vogt, Debra A. Fischer, Gregory W. Henry, Gregory Laughlin, and Jason T. Wright. Seven New Keck Planets Orbiting G and K Dwarfs. *APJ*, 582(1):455–466, January 2003. doi: 10.1086/344570.

[147] Kento Masuda. Very Low Density Planets around Kepler-51 Revealed with Transit Timing Variations and an Anomaly Similar to a Planet-Planet Eclipse Event. *APJ*, 783(1):53, March 2014. doi: 10.1088/0004-637X/783/1/53.

[148] Samuel K. Grunblatt, Daniel Huber, Eric Gaidos, Eric D. Lopez, Thomas Barclay, Ashley Chontos, Evan Sinukoff, Vincent Van Eylen, Andrew W. Howard, and Howard T. Isaacson. Do Close-in Giant Planets Orbiting Evolved Stars Prefer Eccentric Orbits? *APJL*, 861(1):L5, July 2018. doi: 10.3847/2041-8213/aacc67.

[149] Coel Hellier et al. An orbital period of 0.94days for the hot-Jupiter planet WASP-18b. *Nature*, 460(7259):1098–1100, August 2009. doi: 10.1038/nature08245.

[150] Ji-Wei Xie. Transit Timing Variation of Near-resonance Planetary Pairs. II. Confirmation of 30 Planets in 15 Multiple-planet Systems. *APJS*, 210(2):25, February 2014. doi: 10.1088/0067-0049/210/2/25.

[151] Vincent Van Eylen and Simon Albrecht. Eccentricity from Transit Photometry: Small Planets in Kepler Multi-planet Systems Have Low Eccentricities. *APJ*, 808 (2):126, August 2015. doi: 10.1088/0004-637X/808/2/126.

[152] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[153] Diego J. Muñoz, Dong Lai, and Bin Liu. The formation efficiency of close-in planets via lidov-kozai migration: Analytic calculations. *Monthly Notices of the Royal Astronomical Society*, 460, 2016. ISSN 13652966. doi: 10.1093/mnras/stw983.

[154] Alessandro A. Trani, Michiko S. Fujii, and Mario Spera. The keplerian three-body encounter. i. insights on the origin of the s-stars and the g-objects in the galactic center. *The Astrophysical Journal*, 875, 2019. ISSN 15384357. doi: 10.3847/1538-4357/ab0e70.

[155] Alessandro A. Trani, Mario Spera, Nathan W. C. Leigh, and Michiko S. Fujii. The keplerian three-body encounter. ii. comparisons with isolated encounters and impact on gravitational wave merger timescales. *The Astrophysical Journal*, 885, 2019. ISSN 15384357. doi: 10.3847/1538-4357/ab480a.

[156] Rosemary A. Mardling and Sverre J. Aarseth. Tidal interactions in star cluster simulations. *Monthly Notices of the Royal Astronomical Society*, 321, 2001. ISSN 00358711. doi: 10.1046/j.1365-8711.2001.03974.x.

[157] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from Imbalanced Data Sets.* Springer, 2018. doi: 10.1007/978-3-319-98074-4.

[158] Florian Lalande and Alessandro Alberto Trani. Predicting the stability of hierarchical triple systems with convolutional neural networks. *The Astrophysical Journal*, 938, 2022. ISSN 0004-637X. doi: 10.3847/1538-4357/ac8eab.

[159] Jonathan Mushkin and Boaz Katz. A simple random walk model explains the disruption process of hierarchical, eccentric three-body systems. *Monthly Notices of the Royal Astronomical Society*, 498, 2020. ISSN 13652966. doi: 10.1093/mnras/staa2492.

[160] John R. Koza. *Genetic Programming.* The MIT Press, 1992.

[161] Mojtaba Valipour, Maysum Panju, Bowen You, and Ali Ghodsi. Symbolicgpt: A generative transformer model for symbolic regression. In *Preprint Arxiv*, 2021. URL `https://arxiv.org/abs/2106.14131`. Under Review.

[162] Brenden K Petersen, Mikel Landajuela Larma, Terrell N. Mundhenk, Claudio Prata Santiago, Soo Kyung Kim, and Joanne Taery Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021.

[163] Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and Francois Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=GoOuIrDHG_Y`.

[164] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabricio Olivetti de Franca, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. In *Thirty-fifth Conference on NeurIPS Datasets and Benchmarks Track (Round 1)*, 2021.

[165] Yoshitomo Matsubara, Naoya Chiba, Ryo Igarashi, and Yoshitaka Ushiku. Rethinking symbolic regression datasets and benchmarks for scientific discovery. *arXiv preprint arXiv:2206.10540*, 2022.

[166] Silviu Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6, 2020. ISSN 23752548. doi: 10.1126/sciadv.aay2631.

[167] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18, 1989. ISSN 00975397. doi: 10.1137/0218082.

[168] Florian Lalande, Yoshitomo Matsubara, Naoya Chiba, Tatsunori Taniai, Ryo Igarashi, and Yoshitaka Ushiku. A transformer model for symbolic regression towards scientific discovery. In *NeurIPS 2023 AI for Science Workshop*, 2023. URL `https://openreview.net/forum?id=AIfqWNHKjo`.

[169] George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976. ISSN 01621459. URL `http://www.jstor.org/stable/2286841`.

[170] Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, and Peter Battaglia. Rediscovering orbital mechanics with machine learning. *Machine Learning: Science and Technology*, 2023. doi: 10.1088/2632-2153/acfa63.

[171] Miles Cranmer. Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl. *arXiv e-prints*, art. arXiv:2305.01582, May 2023. doi: 10.48550/arXiv.2305.01582.

[172] Konstantin Batygin, Fred C. Adams, Michael E. Brown, and Juliette C. Becker. The planet nine hypothesis. *Physics Reports*, 805:1–53, 2019. ISSN 0370-1573. The planet nine hypothesis.

Appendix A

Automatic Rediscovery of Neptune without calculus

The work presented here has been done under the close supervision of Yasushi Sutō, professor in astrophysics and cosmology at the University of Tōkyō, and in collaboration with Alessandro Alberto Trani, postdoctoral researcher at the University of Tōkyō. Here, we aim at automatically reproducing the historical discovery of Neptune, with minimum analytical assumptions. For this purpose, I develop a Graph Neural Network (GNN) to empirically retrieve the Universal Law of Gravitation of Newton using simulated data. This project has not yet lead to a published article.

A.1 Context

As George Box said in 1976: "All models are wrong, but some are useful" [169]. Nothing guarantees that our world is indeed dictated by a set of exact mathematically expressed laws of physics. Models should instead be considered as approximations to the (almost inaccessible) truth. As such, the forefront of scientific research is always confronted with the dilemma of (i) refining existing laws (new theory), (ii) postulating the existence of unknown components, or (iii) reviewing the errors and the interpretation of observed data.

The history of the Solar System discovery is a great example of such evolution in science. After Newton invented calculus and proposed the universal inverse-square law of gravitation, we could understand the dynamics at play in the Solar System. Staggeringly, the inverse-square law of gravitation seems to be an exact law governing our world, and not a mere approximation of the reality.

The assumption that the inverse-square law of gravitation is exact led to the discovery of Neptune in 1846, after Le Verrier and Adam independently predicted its correct location as an attempt to reconcile the observed motion of Uranus with Newton's law of gravitation.

In our Neptune project, we try to quantify to what extent can we correctly predict the presence of Neptune without the rules of calculus. For example, assuming an intelligent civilization capable of performing arithmetic operations very accurately and quickly (this is what machine learning does), could we infer the presence of Neptune by looking at the inner planets in our Solar System?

In the Solar System, the overwhelmingly dominant force remains the gravitational attraction of the Sun. However, the location of the outer two planets between 1800 and 1840, shown on Figure A.1, maximized the gravitational effect of Neptune onto Uranus.

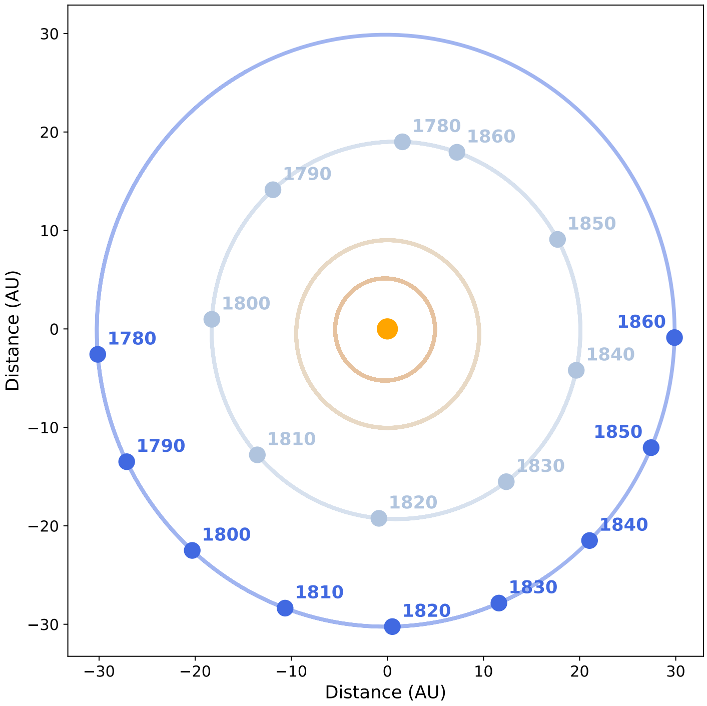


Figure A.1: Orbits and locations of solar planets between 1750 and 1915, one orbital period of Neptune. Location for Uranus and Neptune are labelled every 10 years for the period 1780 - 1860. Uranus was first identified as a planet by Herschel in 1781. Neptune was officially confirmed by Galle in 1846, on the basis of the theoretical predictions by Le Verrier and Adam.

A.2 The effect of Neptune on Uranus

We decompose the total acceleration of Uranus $\vec{a}_{\mathrm{U,tot}}$ due to the Newtonian gravity of objects in the Solar System. This breakdown can be expressed as

$$\vec{a}^{\mathrm{N}}_{\mathrm{U,tot}}(t) \equiv \sum_{j\neq 7}^{8} \frac{Gm_j}{|\vec{r}_j - \vec{r}_{\mathrm{U}}|^3}(\vec{r}_j - \vec{r}_{\mathrm{U}})$$

where indices j denote the Sun (0), Mercury (1), Venus (2), the Earth (3), Mars (4), Jupiter (5), Saturn (6), Uranus (7), and Neptune (8), and m_j and $\vec{r}_j$ denote their mass and position vector respectively. For reference, we compute the acceleration of Uranus caused by Neptune alone

$$\vec{a}^{\mathrm{N}}_{\mathrm{U,Nep}}(t) \equiv \frac{Gm_{\mathrm{Nep}}}{|\vec{r}_{\mathrm{Nep}} - \vec{r}_{\mathrm{U}}|^3}(\vec{r}_{\mathrm{Nep}} - \vec{r}_{\mathrm{U}})$$

Even at its closest approach, the instantaneous acceleration on Uranus caused by Neptune is approximately 5×10^4 times weaker than the Sun's one, making it extremely hard to probe. However, we are not necessarily interested in the instantaneous acceleration caused by Neptune onto Uranus, more rather in its integrated effect, which eventually resulted in the tiny prediction errors that lead to the discovery of Neptune.

The rest of this work involves two parts. The first part, presented in Section A.3 introduces our preliminary sanity checks consisting in retrieving the true orbital parameters for Neptune when assuming Newton's inverse-square law. The second part, in Section A.4, presents the GNN used in this study and shows our latest training results.

A.3 Preliminary results: retrieving Neptune's orbital parameters

If we integrate the motion of the planets in the Solar System assuming the exact inverse-square law of gravitation, it is no surprise that we can reproduce the observed data. Now, if we remove Neptune and integrate once again, we can see a very slight difference in the azimuthal angle of Uranus, caused by the missing gravitational pull of Neptune.

Next, we postulate the existence of Neptune with its correct semi major axis and mass, but with unknown phase. Using a standard grid-search strategy, we search for the phase of Neptune that can explain the observed data by minimizing the azimuthal angle discrepancy between each integration and the observed data. Figure A.2 shows the grid-search results for the phase of Neptune. Four coloured dots have be selected to show the evolution of the difference in the azimuthal angle in the lower panel. The phase that minimizes the RMSE is highlighted by the blue dot, and the real phase for Neptune in our observed data is $\varphi = 129.9\,\mathrm{deg}$.

Following the grid-search of Neptune phase, I relaxed two additional assumption: the semi-major axis and the mass of Neptune. Once again using a grid-search approach (now in 3-d), I was able to simultaneously retrieve the mass, the semi-major axis, and the phase of Neptune by minimizing the discrepancy in the evolution of the azimuthal angle between the observed and the integrated position of Uranus. The true parameters for Neptune in our observed data are given by $m = 1.01 \times 10^{26}\,\mathrm{kg}$, $a = 30.2\,\mathrm{au}$, and $\varphi = 129.9\,\mathrm{deg}$.

A.4 Modeling arbitrary laws of gravity with a GNN

The GNN I developed was inspired from a similar methodology used in a previous study which attempts to rediscover orbital mechanics using a Graph Neural Network [170]. In their work Lemos et al. trained a GNN using NASA Horizons Data, and then fit the learnt relationship using PySR [171], a Python package for Symbolic Regression. Here instead, we do not use symbolic regression and allow for arbitrary analytical form of the gravitational law.

In the general case, the gravitational force caused by body j onto body i may depend on the mass of body j and the relative positions of bodies i and j. In our exercise, we assume mass proportionality, translational invariance, and rotational invariance, such that the gravitational force can be expressed as $\vec{g}(m_j, \vec{x_i}, \vec{x_j}) = m_j \tilde{g}(r) \vec{u_r}$, where $r = \Delta\vec{r} = |\vec{x_i} - \vec{x_j}|$

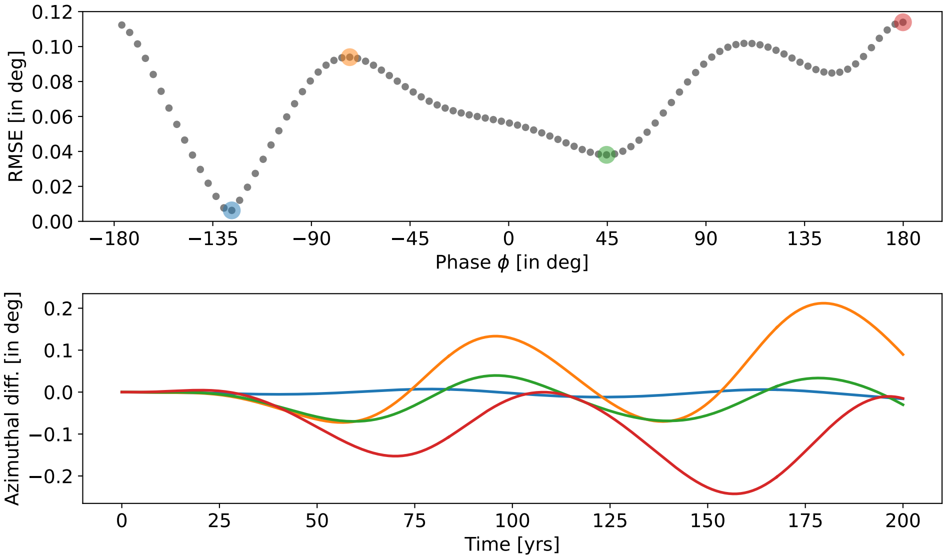


Figure A.2: Grid-search results for the phase of Neptune. Coloured dots correspond to the azimuthal angle differences plotted in the bottom panel. The blue dot yields the minimum error, and the observed phase for Neptune in our observed data is $\varphi = 129.9\,\mathrm{deg}$

is the euclidean distance between body i and body j, and $\vec{u_r}$ is a unit vector from i to j. The aim is now to estimate $\tilde{g}(r)$ by fitting the GNN parameters to simulated data.

Our GNN consists in 9 nodes, each pair of nodes having one non-directional connection (i.e. 36 edges). Each node represents one body in the Solar System, from the Sun to Neptune, and have a single free trainable parameter with correspond to the body's mass. The edges of the GNN represent arbitrary (but shared) relationships between objects. The "edge-function" (which secretly represent the Law of Gravitation in our case) is freely modeled by a standard multi-layer perceptron (MLP).

We simulate training data using TSUNAMI. During training, we remove data on Neptune and train using only the Sun and the innermost seven other planets. The GNN is provided the position of each body in Euclidean coordinates, and is tasked to predict the observed acceleration, computed by taking twice the finite difference of the position. Finally, we also assume Newton's Second Law of Physics $\sum \vec{F} = m\vec{a}$ such that the GNN is now tasked to estimate $\sum \vec{F}$, which is a proxy for the observed acceleration $\vec{a}$.

Figure A.3 shows the training results after convergence of our GNN. The blue thick line shows the estimated law of gravitation resulting from the "edge-function" of the GNN, while the real inverse-square law of gravitation is shown in orange. Training data have pair-wise distances ranging from approximately $2.5 \times 10^{-1}\,\mathrm{au}$ to $30\,\mathrm{au}$. This corresponds to the range where the MLP of the GNN is close to the inverse-square law.

Although the GNN-learnt law of gravitation is very close to the inverse-square law, it is still not precise enough to predict the evolution of the Solar System even for integrations of few decades, due to the accumulation of small imprecisions over time. For that reason, we cannot adopt the same strategy as before because the integration of the system will always result in very high azimuthal angle errors regardless of the orbital parameters for Neptune because of the inaccurate law of gravitation.

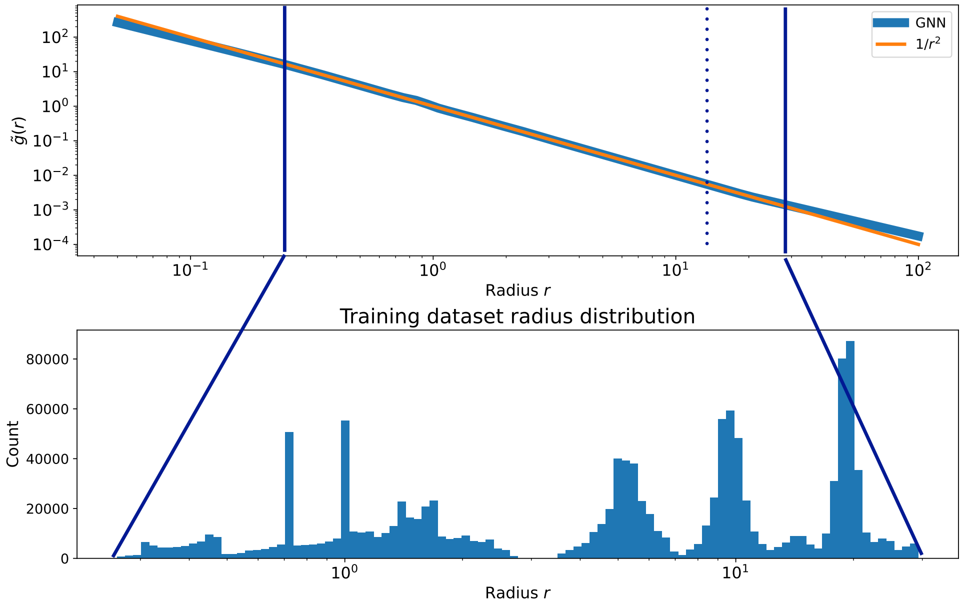


Figure A.3: Approximated law of gravitation by the GNN. Note the log-log scale. The approximation of the GNN looks very close to the real inverse-square law of gravitation over the range of distances represented in the training dataset. This law does not generalize well for out-of-domain distances. The GNN simultaneously learns the law of gravitation and the planet masses.

To try to circumvent this problem, I implemented a new feature to the GNN model which allows to assume a power law for the law of gravitation (instead of a completely arbitrary law over the range of possible distances). With this new framework, the GNN now has a single parameter in its “edge-’function”, corresponding to the power law exponent. Training following the same strategy has been performed 50 times to account for variability, and the results are shown in Figure A.4.

Training results show that the law of gravitation is always systematically underestimated, with an exponent typically between -1.9996 and -1.9997, instead of the expected -2.0 for the inverse-square law. We do not know what is causing this consistent underestimation, and further analysis will be needed here. Interestingly, it might be linked with the systematic underestimation of the Sun’s mass. because the Sun is by far the most massive body in the Solar System, and therefore the body which dictates the general motion of all planets, there is indeed a sort of degeneracy between the Sun’s mass and the strength of the law of Gravity: if the Sun’s mass is lower, the gravity’s strength has to be slightly lower as well to explain for the same observed acceleration.

A.5 Temporary conclusion and Future work

We are still left with the question whether AI can automatically rediscover Neptune without the rules of calculus. The current state of this project suggests that the very exact Law of Gravitation by Newton is necessary to computationally probe the effect of

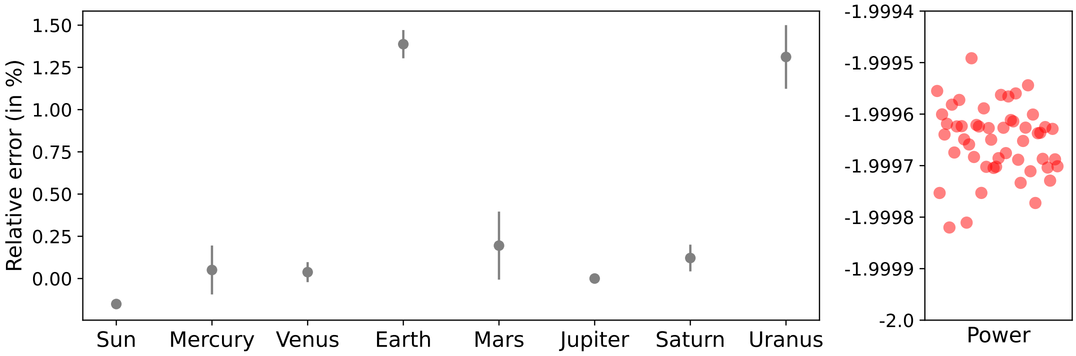


Figure A.4: Fit of the law of gravitation assuming power law. The masses of the planets and the standard deviation (computed over 50 trials) are shown on the left panel, while the optimal power-law exponent is shown on the right panel. These results indicates a systematic underestimation of the law of gravity.

Neptune on Uranus.

By allowing the law of gravitation to depend on the distance, the MLP of the GNN estimates something very close to the inverse-square law, but still not precise enough for our purpose. When integrating the motion of the planets using the GNN-learnt law, small imprecisions accumulate over time and rapidly lead to inaccurate positions. It is also worth noting that, the GNN-learnt law of gravitation is not 100% arbitrary in fact, because using an artificial neural network still implies some assumptions by design: the continuously differentiable activation functions leading to overly smooth outputs, and the architecture of the neural network itself is a working hypothesis as well.

Future steps involve quantifying the relative precision required to detect the presence of Neptune after integration over few decades. Even if the GNN-learnt law of gravitation is slightly incorrect, are we still able to probe inconsistencies in the apparent movement of Uranus and automatically detect Neptune. Although it appears like a scholar exercise, the results of this work may provide important insights for the astrophysics researchers using Machine Learning tools for the hunt of Planet Nine within our Solar System [172].