RAS Techniques and Instruments

RASTAI **3,** 288–314 (2024)
Advance Access publication 2024 June 11

https://doi.org/10.1093/rasti/rzae020

Exoplanet host star classification: multi-objective optimization of incomplete stellar abundance data

Miguel A. Zammit[1,2]★ Josef Borg[1,3] and Kristian Zarb Adami[1,2,4,5]

[1] *Institute of Space Sciences and Astronomy, University of Malta, Msida, MSD 2080, Malta*
[2] *Department of Physics, University of Malta, Msida, MSD 2080, Malta*
[3] *Faculty of Health Sciences, University of Malta, Msida, MSD 2080, Malta*
[4] *Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK*
[5] *Osservatorio Astrofisico di Catania, Via S. Sofia 78, 95123, Catania, Italy*

Accepted 2024 June 3. in original form 2024 May 27

ABSTRACT

The presence of a planetary companion around its host star has been repeatedly linked with stellar properties, affecting the likelihood of substellar object formation and stability in the protoplanetary disc, thus presenting a key challenge in exoplanet science. Furthermore, abundance and stellar parameter data sets tend to be incomplete, which limits the ability to infer distributional characteristics harnessing the entire data set. This work aims to develop a methodology using machine learning (ML) and multi-objective optimization for reliable imputation for subsequent comparison tests and host star recommendation. It integrates fuzzy clustering for imputation and ML classification of hosts and comparison stars into an evolutionary multi-objective optimization algorithm. We test several candidates for the classification model, starting with a binary classification for giant planet hosts. Upon confirmation that the eXtreme Gradient Boosting algorithm provides the best performance, we interpret the performance of both the imputation and classification modules for binary classification. The model is extended to handle multilabel classification for low-mass planets and planet multiplicity. Constraints on the model's use and feature/sample selection are given, outlining strengths and limitations. We conclude that the careful use of this technique for host star recommendation will be an asset to future missions and the compilation of necessary target lists.

Key words: Machine Learning – Algorithms – Data Methods – Astronomical Data Bases: Miscellaneous – Planetary Systems: Exoplanets – Stars: Abundances.

1. INTRODUCTION

The advent of large and diverse data sets of exoplanets and their stellar hosts, particularly due to the *Kepler* and *TESS* space missions (Ricker et al. 2014; Borucki 2016) and ground-based surveys such as WASP, HATNet, and KELT (Bakos et al. 2002; Pollacco et al. 2006; Pepper et al. 2007), has made possible the exploration of correlations within the data to further inform planet formation theories and scenarios. Several bodies of work have attempted to map distributional statistics of stellar and dynamical parameters with planet occurrence, primarily using samples which are somewhat limited in size, thus inhibiting the generalizability of the results (e.g. Johnson et al. 2010; Dai et al. 2021). This limitation is especially pronounced when incorporating elemental abundances of the stellar hosts, the measurement of which tends to be time-consuming and dependent on the nature of the star. None the less, assessing chemical and physical correlations between exoplanet host stars and the occurrence and type of planetary companions provides a promising avenue of research as the data sets become larger (and more complete) and application of the powerful predictive capabilities of machine learning (ML) algorithms becomes feasible.

One of the first correlations in stellar host samples which remains consistently present in the majority of the relevant literature is the chemical correlation between stellar metallicities and planetary occurrence (Gonzalez 1997; Gonzalez & Laws 2000; Reid 2002; Petigura et al. 2018). This is especially true for giant planet occurrence, for which it was noted that Jovian hosts in the solar neighbourhood tend to be metal-rich, implying that the stellar metallicity is linked to the efficacy of planetary formation (Santos et al. 2005). Progress in exploring this correlation was inevitably linked to the availability of larger spectroscopic samples and the frequency of detected planets. Naturally due to transit photometry and Doppler spectroscopy being more sensitive to larger planet-star mass and radius ratios, the population of detected giant planets dominated over the lower mass regime (Zhu & Dong 2021). Hence, all correlations derived for giant planets were based on larger samples and thus making the results, at least statistically speaking, more reliable. None the less, several bodies of work in the literature have investigated the possible cause of the correlation. Selection effects have been suggested (e.g. Sozzetti 2004; Paulson & Yelda 2006) as these have discrepancies between magnitude and volume-limited surveys (e.g. Fischer & Valenti 2005) and even the nature of the metallicity metric itself (e.g. Gonzalez 2014). For giant planets, however, even when a number of these biases were addressed, the correlation still persisted (Gonzalez 2014). For low-mass planets, there are indications of a metallicity cliff, such

★ E-mail: mzamm33@um.edu.mt

that a lower limit can be imposed on the metallicity required for terrestrial planet formation (see e.g. Boley & Christiansen 2023), and compositional links between terrestrial planets and their hosts have also been found (Adibekyan et al. 2021).

As the wealth of host star samples grew and became more diverse, several constraints of the parameter space have been found, such that the nature, strength, and persistence of the metallicity–occurrence correlation change with both stellar and planetary parameters. The correlation is not found to be valid at intermediate metallicities (Haywood 2009), although the work in Lineweaver (2001) suggests that terrestrial planet formation is facilitated in these cases. In Santos et al. (2017), hosts of $> 4M_J$ giant planets tend to be more massive and metal-poor. For Neptune planets, the correlation for metal-rich hosts was not present in Sousa et al. (2008), as was the case in Buchhave & Latham (2015) for hosts for low-mass planets. Furthermore, the correlation weakens or disappears entirely for certain stellar types. Maldonado, Villaver & Eiroa (2013) show that giant stars which host planets are not preferentially metal-rich, and Mann et al. (2013) find that the late K and M dwarfs in their sample show no discrepancies between the two subsets. This being said, the M dwarf host stars in Johnson & Apps (2009) sample are all systematically metal-rich, and the Ida & Lin (2005) sample suggests that hot Neptunes may be more abundant in this case.

Several other abundance correlations have also been explored in the literature. Higher rates of lithium depletion in stellar atmospheres have been linked to fast rotators, an effect which can be induced by the presence of a planetary companion (Delgado Mena et al. 2014, 2015). Samples such as that of Israelian et al. (2004) show evidence for this discrepancy at its strongest for solar-type effective temperatures, labelling it as 'significantly different' for 5600–5850 K and not significant for the higher range of 5850–6350 K. Israelian et al. (2004) did attempt to find correlation between the Li abundance of the parent stars and various parameters of the planetary companions, yet did not find any strong correlation in their sample. There has also been a lot of work done in exploring refractory and volatile abundances, especially due to their known strong influence on planet formation. Consistently significant trends have thus far remained elusive, suggesting that whilst both sets of elements are important in determining planetary formation scenarios, they are not intrinsic markers within the star to identify hosts (Bodaghee et al. 2003; Gilli et al. 2006; Perryman 2018). Some relative abundance, such as Mg/Si ratios, have been linked to facilitate the formation of low-mass planets. The work presented in Mah & Bitsch (2023) suggests that low Mg/Si ratios can potentially aid in the formation of super-Mercuries. In the case of r-process and s-process elements, whilst links have been more tentative than the metallicity correlation, some samples have shown differences between the two subsets (Bond et al. 2008; Perryman 2018).

Besides searching for independent relationships discriminating between hosts and comparison stars, some work has probed which correlations become more important in the metal-poor regime (Adibekyan et al. 2012a, b). α-element abundances in the *Kepler* and HARPS samples used in Adibekyan et al. (2012a, b) show a clear enhancement for host stars which are metal-poor. Adibekyan et al. (2012a) add that this is particularly relevant for low-mass planets, out of which most of the sample was composed. At the low-iron abundance regime, it seems likely that other metals contribute to facilitating planet formation (Adibekyan et al. 2012b).

Besides chemical correlations, there are several physical and galactic parameters which are expected to correlate with not only the presence of a planetary companion around a star but also constrain formation scenarios for which class of planets are likelier to form. The occurrence rate by planetary type has been shown to vary dependent on stellar type, and as mentioned previously affects the strength of the chemical correlations we should expect. Dai et al. (2021) utilized the *Gaia–Kepler* Stellar Properties Catalog to link planet occurrence rate with stellar relative velocities, finding the high-V stars have lower occurrences of super-Earths and sub-Neptunes, whilst having a higher occurrence rate for sub-Earths. They also show that high-V stars have a lower occurrence of hot Jupiters and a slightly higher one for warm or cold Jupiters. In addition, stellar rotation and angular momentum have both been investigated for their correlations with planetary occurrence, and certain models and samples did show some discrepancies (see e.g. Gonzalez 2008, 2015; Alves, Do Nascimento & De Medeiros 2010; Lanza 2010). Due to its link with lithium depletion, stellar rotation correlations are one example of physical stellar characteristics that can manifest themselves as chemical correlations. Galactic location during formation has also been suggested to play a role in aiding planet formation. In Haywood (2008, 2009), the variation in the strength of the metallicity–occurrence correlation was argued to suggest that the mechanism was galactic, rather than linked to formation. Galactic location and thin/thick disc location would, in this scenario, be the contributing factors. It should be noted however that once a refractory index was used instead of metallicity in Gonzalez (2009, 2014), these dependencies were no longer significant. None the less, while the current standing in the literature tends to favour primordial chemical composition models over galactic location for the source of these correlations, work on α-element abundances does suggest that disc location and U, V, and W velocities can hold promise (Adibekyan et al. 2012a, b).

The applicability of ML algorithms to this problem has seen several previous works conducted in order to exploit any correlations to develop a recommendation algorithm for potential host stars. Hinkel et al. (2019) use their Hypatia Catalog (Hinkel et al. 2014) to train an eXtreme Gradient Boosting (XGBoost) algorithm to recommend potential giant exoplanet host stars, detecting significant trends which suggest that HIP62345, HIP71803, and HIP10278 host long-period giant planet companions. Inspired by this work, we have developed a spectral classifier in Zammit & Zarb Adami (2023, 2024) which approaches the problem from a machine vision standpoint. Rather than taking elemental abundance data, the model takes a high-resolution spectral input and labels each spectrum as either a comparison star or a gas-giant host. Both architectures of the model achieve relatively strong generalization scores, with the accuracies on the test set both higher than 94 per cent. However, overfitting could not be entirely alleviated regardless of all regularization techniques applied to the model. The main cause of this is expected to be two-fold: the training regime for the model requires a larger data sample, and the input being spectral will in of itself hit a performance ceiling. In our concluding remarks in Zammit & Zarb Adami (2024), we allude to the possibility that rethinking the input design might serve as a solution to increasing stability in model performance.

If the choice is taken to move from a spectral input to a set of abundance and physical features, an immediate challenge arises due to high degree of incompleteness in the data set. Since abundance measurements are heavily reliant on several astrophysical, observational, and instrumental systematics, each sample in the data will contain null elements. Since ML algorithms tend to assume completeness, null elements need to be handled during the pre-processing stage. There are several techniques in which this is done. The simplest method would be to simply omit incomplete samples, provided they are relatively uncommon within the data set. Hinkel et al. (2019) incorporate the omission of missing features for a specific sample in the mechanism for fitting their decision trees, such that they avoid a whole omission of a sample, which works well for specific algorithms and is particularly effective in their work. However, the application of several different classification algorithms and neural networks in such a way that their optimization follows the typical ML methodologies would require techniques for imputing the missing features and completing the data set. Many imputation techniques have been developed, and whilst they vary substantially in complexity, they are all inherently offline processes carried out before fitting and training of the ML model (Khorshidi, Kirley & Aickelin 2020). Hence before training, no information other than preliminary data explanatory analysis can inform the imputation method in determining which method directly contributes to better predictive performance by the model. This led to the methodology by Khorshidi et al. (2020), in which an online approach was recommended for ML with incomplete data through the application of multi-objective optimization (MOO). Rather than treating the imputation and classification as two entirely independent endeavours, incorporating both within a MOO evolutionary algorithm explores the trade-off between the two methods to find a set of non-dominated solutions, i.e. a set of hyperparameters which provide statistically reliable imputation and strong and consistent generalization from the ML model. In their original work, Khorshidi et al. apply a fuzzy clustering technique for imputation and a support vector machine (SVM) as the classification algorithm. The use of the former in providing an imputation technique which is not only online but also a ‘smarter way to formulate solutions for finding optimal solutions in comparison with using all missing values as decision variables’, was the particular reason why we decided to apply this methodology to our work.

In this work, we propose the use of the overarching methodology in Khorshidi et al. (2020) to develop an exoplanet host classifier applied to a large incomplete stellar abundance and parameter data set. Rather than to build an entirely generalizable classifier to use external to this sample, we focus on using the predictive strength of ML to build a complete data set on which it would then be possible to explore the correlations between occurrence and stellar parameters and chemistry. Hence, the long-term objectives of this work align with that of Hinkel et al. (2019), and we, in fact, make use of the same stellar data set, the Hypatia Catalog. Where our work diverges, besides the obvious difference in a wholly different methodology for applying our algorithms, is in the fact that we aim to train an imputation method to obtain a statistically reliable complete data set. This can potentially allow for any subtle correlations previously hidden by a low completion rate for certain features, to be more present once the imputation is applied. Furthermore, since fuzzy clustering considers the distributional characteristics over the entire dimensionality of the data, it can theoretically identify certain trends in the data which may prove interesting to explore, trends which would otherwise be masked by a low number of non-null samples. It may also be useful not only for comparative tests between samples to uncover trends, but rather to use as a recommendation method for data instances which appear to be clustered with hosts rather than comparison stars. Whilst a substantial caveat in its application to other planetary classes lies in the fact that sensitivity biases persist in making detection more rare, we extend our classification labels to not only include giant planet hosts but also incorporate labels for low-mass planet hosts and planet multiplicity, changing the problem from a binary classification task to one which is multilabel. Since the data set contains a substantial number of stars which fall into either one of these three labels, it was deemed fruitful to develop a multilabel pipeline to explore both the classification performance and the distributional statistics in the imputed data set. The *Kepler* and *TESS* samples have led to a large increase in rocky planet detection (Lissauer, Dawson & Tremaine 2014; Brady & Bean 2022), and have thus led to an influx of sample correlation analytics exploring any discrepancies between host and comparison stars. Furthermore, the question of whether planet multiplicity is itself linked to host star properties may be an interesting avenue to explore, either as a proxy for dynamical stability or as an indicator of the available chemical budget and protoplanetary disc properties to facilitate/inhibit planet formation. Hence, transforming the problem from being a two-dimensional (2D; imputation and giant planet classification) to one which is 4D (imputation and giant planet, and low-mass-planet and plane-multiplicity classification) can reveal further distributional characteristics within the sample once the data set is complete.

In Section 2, we describe our entire methodology, starting from acquiring and preparing the data set in all its several forms, and moving onto a high-level description of how the MOO genetic algorithm (GA) from Khorshidi et al. (2020) was adapted for our specific task. In Section 3, we delve into more detail on the specific modules within the GA, describing the imputation and classification modules, as well as the crossover and mutation operations implemented to create the next generation of chromosomes within the algorithm. In Section 4, we present the first set of preliminary results used to validate our system and choose which classification model shows greater promise for our particular data set. In Sections 5 and 6, we explore all classification and imputation metrics of the system when framed as a binary classification task and multilabel classification task, respectively. Finally, Section 7 will cross-examine all results and present our final discussion points, with concluding remarks on potential avenues for future work.

2. DATA AND METHODOLOGY

Every endeavour in ML is heavily dependent on the quality, more so than the quantity, of its data set. Therefore, ensuring that the curation and preparation of all samples to suit that specific task, before attempting any training or fitting, is paramount. In this section, we will describe our methodology in how we first prepare our full data set, then in how we define our variation in both features and samples such that we can then test the performance of both the imputation and classification modules over different scenarios.

2.1 The Hypatia Catalog

The Hypatia Catalog[1] is a spectroscopic abundance data set for FGKM-stars in the solar neighbourhood, with particular focus placed on compiling a comparative sample for exoplanet host stars. Comparison stars within the sample have been volume-limited to within 500 pc of the sun, whilst all exoplanet host stars for which there is abundance data have been included. The data set has been carefully curated from over 200 literature sources, totalling 9982 stars at the point at which the data set was acquired. Stellar properties and planetary companion parameters are also provided, ensuring that any correlation analysis conducted can explore chemical, physical, and galactic connections. For a full thorough description of the data set, the reader is directed to Hinkel et al. (2014) in which the first iteration of the data set was published.

We acquired the data set in May 2023, and thus use all stars included up to that date. The full data set amassed 9982 stars with varying feature completion rates. The solar normalization method

[1] https://www.hypatiacatalog.com/

used is that described in Lodders, Palme & Gail (2009), which is generally the default setting for the Hypatia API. In cases where the elemental ratio for a particular star is found in multiple catalogues, the median value was taken. As will be explained in Section 2.2, target labelling for the classification tasks was verified through cross-referencing with other exoplanet archives, and not solely reliant on the values within the catalogue itself. This ensured that labelling was as confident as possible. The selection of features from the entire data set to include in both the imputation and classification modules also follows a specific methodology influenced by the aims and motivations for this work, as will be explained in Section 2.3.

2.2 Target labelling and verification

For a data set used in a ML classification task, ensuring that target labels are accurately assigned and unambiguous is essential. After all, without correct labelling, the transformation function which the model will attempt to converge to relies entirely on the distributional characteristics of the training set imposed on the input features by the labels.

We test two variations of the classification task, one of which is a binary classification of giant planet hosts, and the second is a multilabel classification for hosts of giant planets, low-mass planets, and planet multiplicity. This naturally requires the use of planetary parameters and pre-defining thresholds, as the following paragraphs will explain.

For full confidence in ensuring our labelling used up-to-date corroborated values for all samples in our data set, the planetary parameters in the Hypatia data set were cross-referenced with the NASA Exoplanet Archive[2] and the exoplanet.eu[3] data base. This was done by acquiring the archival data for each sample with PyAstronomy's AstroLib[4] routines and verifying that all planetary parameters are updated, replacing the values otherwise.

The actual process of determining the labels for each sample can be divided into conditionals depending on which planetary parameters are available, iterating over the number of companions to serially update the final set of labels. Due to its more indicative role in determining planetary class the first conditional attempts to label the entry dependent on the planetary mass values. If the mass is not fully constrained and a minimum value is available through radial velocity measurements, this is used in its stead. The mass classification scheme is based on the 'minimum' giant planet mass defined as $0.1M_{\rm J}$ in Clanton & Gaudi (2014). They argue that by the steep mass function implied by microlensing surveys, this value presents a likelihood for planets more massive than this cut-off to be composed of >50 per cent H and He by mass, provided that the heavy element content is not $\gg$ 10 per cent and the protoplanetary disc was not very massive (Clanton & Gaudi 2014; Perryman 2018). The choice to employ a single cut-off separating low-mass and giant planets, rather than incorporating a 'Neptunian' threshold, was to avoid incorporating a second potential source for astrophysical biases in our labelling and minimize the amount of ambiguity in the data set. An upper limit of $13M_{\rm J}$ was also present to exclude any brown dwarfs. If no mass is constrained, then the planetary radius is used to determine if the label requires an update. The radius classification scheme of Borucki et al. (2011) was used as a cut-off point for giant planets. It is important to note that whilst hierarchical importance was placed in prioritizing masses over radii, all schemes were checked and if any conflicts arose they were to be flagged for manual verification with the literature. With the classification schemes in place, for each sample, we set the giant planet host and low-mass planet host labels as 1 if at least one of their respective type of planet is present. If the host has more than one companion, the planet multiplicity label is also set to 1.

The multiplicity label does however require some justification. As mentioned by Sandford, Kipping & Collins (2019), strong observational biases impact the current distributional characteristics of multiple-planet hosts, with low-multiplicity systems dominating the sample, especially in the *Kepler* sample. Therefore, strong caution should be taken when including multiplicity as a feature label. Nevertheless, since the observational biases mainly arise from the intrinsic limitations of the transit and radial velocity methods, the upcoming exploration of other detection techniques in future surveys may, to an extent, alleviate the strength of the present bias. It further allows for the opportunity to constrain performance with the inclusion of a known biased label such that we can evaluate its impact on imputation. As this work is a preliminary exploration of the implementation of this method, we choose to include it in the multilabel classification run to fully explore the optimization process and signal injection through multiple labels with varying degrees of instance balance. Hence, while we do not necessarily recommend its inclusion when the methodology in this work is applied, we provide constraints on its usage.

Labelling depends not only on the parameter thresholds but the availability of the parameters themselves. As it is heavily reliant on which detection technique/s are viable for that particular system, planetary radii and masses (and by extension the bulk densities) are not always simultaneously constrained. Moreover, masses can in some cases only have a lower bound value. Therefore, it is important to note that a caveat in the labelling comes about from the fact that some samples will be more confident than others, and combined with the regular detection of new planetary companions, the nature of the data set will be dynamic and require regular updates.

2.3 Feature selection and preparation

Feature selection will have a significant effect on every ML task. Hence, choosing the correct list of parameters to include as input features fed into the model requires thought and consultation to the literature. Since Hypatia has an exhaustive list of chemical, physical, and galactic parameters, the multidimensional representation of each star should be expected to yield results in which discriminating factors between all classes can be highlighted. However, due to the varying degrees of completion across the features, care needs to be taken not to select features with too little of a constrained sample size, hence placing a burden on the imputation stage which will at best confuse and at worst entirely derail the optimization. At the same time, the fact that the aim of this work is to develop a model to help explore the correlations with as little manipulation of the data as possible beyond the feature variation explained below, we avoid implementing feature selection techniques at this stage other than a few simple selection rules based on the feature completion rates. Feature selection will be one of the primary avenues for future work, as explained in Section 7.

Focusing on the chemical feature selection, it was important that the resultant completion rate and its effect on the reliability of the imputation are kept in mind. As the target labels are included in the imputation stage, a lower completion rate will result in more bleeding of information during imputation, which will more strongly

[2] https://exoplanetarchive.ipac.caltech.edu
[3] https://exoplanet.eu
[4] https://pyastronomy.readthedocs.io/en/latest/pyaslDoc/pyasl.html

bias the classification scores. This in of itself is an acceptable and intentional consequence of the methodology adopted,[5] yet should be controlled and not excessive in which point the systematics become overwhelming to the inherent signal being investigated. Therefore, different thresholds were adopted for different tests and cross-examinations presented in this work.

For the results used to select the classifier implemented in the MOO algorithm in Section 4, we use a higher completion rate threshold than what is used for the results in Sections 5 and 6, to ensure that the imputation is less reliant on the influence of label imputation such that it allows for a less-biased interpretation of the strength of generalization seen by the classification module. To this end, we select that all chemical features have above 35 per cent completion. An ideal cut-off completeness rate for features would be 50 per cent, ensuring that the majority will inform the imputation on the minority. However, as only 16 elements satisfy this criterion (Fe, Ti, Ca, Si, Ni, Mg, Na, Al, Cr, V, Mn, O, C, Co, Y, and BaII), a full chemical picture would be missing. Furthermore, for comparison tests with the Hinkel et al. (2019) study, not all elements would have been included. Hence, we dropped the cut-off point by a further 15 per cent to amass 28 chemical features, which would include all the elements in the Hinkel et al. study. A regrettable exclusion from the list of chosen abundances at the model selection stage is the lithium abundance. Li has a low completion rate of 33.60 per cent, unsurprisingly so for its low number of clear spectroscopic lines in the visible range (only Li I at 670.8 nm; Perryman 2018). As mentioned in Section 1, lithium depletion tends to be one of the strongest correlations with giant planet occurrence, and hence should ideally be an important feature for this particular task. However, with its low completion rate, one would risk severely biasing the imputation module to produce a complete data set which is not statistically representative of the true expected population, and would therefore skew the interpretation of performance of the classification module.

For the main implementations presented for the binary label classification in Section 5 and the multilabel classification in Section 6, it became apparent that several elements which the literature highlights as potentially important are omitted, such as Li and certain species of lithophiles. Hence we extend the completion rate threshold twice to create two variants of the data set, one at 25 per cent, and one at 9 per cent. As lithium would be included in the former, this is then adopted as the 'prototype' chemical feature set to be used in generating the other variants listed below. The 9 per cent threshold set was not perpetuated in the other variants due to concerns about the aforementioned imputation pollution, and hence care should be taken when evaluating the classification metrics down the line. The reasoning behind the selection of a 25 per cent threshold is twofold. First, and the main reason for selecting a lower value than the previous 35 per cent, was to include Li and N[6] as features. Second, upon inspection of the completion rates of all features, a sharp drop-off can be seen around 25 per cent completion rate, with the next-most complete feature scoring 15.98 per cent. Therefore, it appeared that such a choice would include sufficiently complete features. The choice of the 9 per cent completion rate comes from the motivation to include primary and secondary abundances for volatile and refractory elements which have been reported to be important for terrestrial formation (K, SiII, CaII, and VII), as well as r-process elements such as Eu (e.g. Bond et al. 2008), which have, again, been investigated for correlations (Hinkel et al. 2019).

Table 1. The selected features for our 'full feature' data set. The total number of features is 51, with 40 chemical, 7 physical, and 4 galactic features. Chemical elements noted in italics fall within the 25 per cent completion rate, whilst those which are also underlined fall under the 9 per cent completion rate.

Full selected data set features					
		Chemical features			
Fe	C	O	Na	Mg	Al
Si	S	Ca	Sc	ScII	Ti
TiII	V	Cr	Mn	Co	Ni
Cu	Zn	Sr	Y	Zr	BaII
LaII	CeII	NdII	EuII	*Li*	*N*
SmII	*ZrII*	*YII*	*CrII*	*PrII*	*SiII*
VII	*Eu*	*CaII*	*K*		
		Physical features			
T_{eff}	$M_{\star}$	$R_{\star}$	$\mathrm{Dist}_{\star}$	m_V	$B - V$
$\log g$					
		Galactic features			
U-vel	V-vel	W-vel	Disc		

Shifting onto the stellar parameters included in the feature list in Table 1, it was important to provide as much explanatory information as possible. The effective temperature, stellar mass and radius, log g, magnitude, and $B - V$ colour were all included. Additionally, the distance in parsec was also selected as a representation for whether or not the star lies within the solar neighbourhood. Finally, we include the thin/thick disc location, and the U, V, and W velocities as a measure for galactic history, since the literature highlights some possible sources of correlations which are galactic in nature (Haywood 2009; Gonzalez 2014). It is important to remember that a higher quality of information regarding the physics and behaviour of the star at this stage will not only help to directly affect model performance during the classification stage but will dimensionally aid the clustering to detect more nuances between different subgroups of stars, allowing for potentially more representative imputation of missing chemical features. The full list of selected features is given in Table 1.

As mentioned in Section 1, observing the change in performance as the feature selection was manipulated, was an important aim of this work. We wanted to explore the importance of specific categories of features not only for the classification, but also for the effect it would have on the final imputed data sets. Hence we select a total of 7 different variations, which in reality can be split into four main variants with one being split further into four subvariations:

(i) Full selected feature data set (25 per cent completion; 45 features): 34 chemical, 7 physical, and 4 galactic

(ii) Full selected feature data set (9 per cent completion; 51 features): 40 chemical, 7 physical, and 4 galactic

(iii) Selected chemical features (25 per cent completion; 34 features): 34 chemical

(iv) No physical features (25 per cent completion; 38 features): 34 chemical and 4 galactic

(v) Selected physical features with specific chemical groups (25 per cent completion)

> (a) Volatiles, lithophiles, siderophiles, Fe (27 features): 20 chemical features and 7 physical
>
> (b) Volatiles, lithophiles, Fe (23 features): 16 chemical features and 7 physical

[5] The reader is reminded that the aim of this work is to build a tool to explore a data set imputed *with* the knowledge of planetary companions, and to evaluate the resultant distributional characteristics.

[6] A compositional tracer for both giant and terrestrial bodies.

(c) Lithophiles, siderophiles, Fe (25 features): 18 chemical features and 7 physical

(d) All selected chemical features (41 features): 34 chemical features and 7 physical

The first four levels of variations were done to check whether omitting specific categories of features (variants 1 and 2 with both physical and galactic features; variant 3 without physical and galactic features; and variant 4 without physical features) would manifest in a change in the performance of the optimizer and classifier. Then, with the fifth set of variants, we were influenced by Hinkel et al. (2019) to observe whether specific chemical groups generally viewed as important for planet formation, combined with physical features would perform better than the full data set, highlighting which chemical species tend to attribute to better model performance. As mentioned in Hinkel et al. (2019), volatiles (C, O) tend to be important for planet atmospheres, and lithophiles (Na, Mg, Al, Si, Ca, Sc, Ti, V, Mn, and Y) are vital for core accretion models, not only playing a pivotal role in giant planet formation but also being a primary component for terrestrial planet formation. Finally, siderophiles (Cr, Co, and Ni) being heavier iron-peak elements, are likely to be present in planetary cores for Earth-like planets, so their abundances should prove fruitful to test. The data set completeness statistics for all feature variants are provided in Table 2. As can be seen in their completion rates, the majority of the variants are missing a substantial portion of their features, highlighting the importance of ensuring imputation is done multidimensionally to maximize the use of information present in the data set. The three variants from which we aim to comparatively assess our results with the work in Hinkel et al. (2014) are relatively far more complete, which should provide greater strength in the imputation stage.

2.4 Sample variation

As mentioned in Section 1 and repeatedly stressed in the literature, the nature and strength of the correlations will vary with sample selection and the stellar population considered. Hence, whilst it is worthwhile and important to observe whether these nuances are captured by the model, varying the data set by being selective in which samples to include will have effects on the performance of both imputation and classification modules. An overview of the data set statistics for all variants of the data set are given in Table 3.

The first modification was based on the findings in work such as that found in Maldonado et al. (2013) that argues that the metallicity–occurrence correlation does not apply to giant stars. Hence we use the spectral type feature in the Hypatia data set to omit the giant star sample from the data set, such that any potential pollution of the data set from uncorrelated samples would not influence the classification module, and more importantly, the imputation of missing features.

The second modification is on the opposite end of the stellar mass regime. Due to limits imposed on giant planet occurrence by formation models and the metallicity cliff (Boley & Christiansen 2023), as well as the diminished influence of the metallicity correlation (Mann et al. 2013), removing low-mass stars should provide similar benefits as the first modification, whilst highlighting some further influence on the correlations for lower mass planets. We select a limit of 0.5 $\mathrm{M}_\odot$ as suggested by Burn et al. (2021) as the point above which giant planets begin emerging in their formation models. This will be mainly used to evaluate the performance on the giant planet host label, focusing on the emergence of the metallicity–occurrence correlation.

The third and final modification is the intersection of the previous two in that both low-mass and giant stars were excluded from the sample. This essentially limits the sample to main-sequence stars excluding low-mass stars.

2.5 Application of MOO

The application of MOO arises in cases where two or more potentially conflicting objective functions require minimization, such that a set of optimal solutions can be found. These solutions, called Pareto-optimal solutions, can then allow for the exploration of the trade-off between objective functions in the fitness/objective space. Emmerich & Deutz (2018) define the optimization problem as follows:

Given a decision space χ and m objective functions

$$f_1 : \mathcal{X} \rightarrow \mathbb{R}, \quad f_2 : \mathcal{X} \rightarrow \mathbb{R}, \quad \cdots \quad f_\mathrm{m} : \mathcal{X} \rightarrow \mathbb{R}$$

$$\text{Minimize} \quad f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_\mathrm{m}(\mathbf{x}) \quad \mathbf{x} \in \mathcal{X}. \tag{1}$$

A popular algorithm used for multi-objective problems, particularly known for its relatively fast computational time aptitude for resulting in a diverse Pareto-optimal front, is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) developed by Deb et al. (2002). For a population of size N, objective algorithm's non-dominated sorting approach reduced the computational complexity from $O(mN^3)$ to $O(mN^2)$, whilst also applying an elitism strategy to ensure the high-fitness chromosomes are carried forward to the next generation. Its functionality is similar to typical GAs in concept. NSGA-II randomly generates a chromosome population as its first generation. All m fitness/objective functions are computed for each chromosome, and the chromosomes are then sorted by Pareto dominance.

From Emmerich & Deutz (2018), given two vectors $\mathbf{y}^{(1)}, \mathbf{y}^{(2)} \in \mathbb{R}^m$, the former Pareto dominates the latter if and only if

$$\begin{aligned} &\forall i \in \{1, \cdots, m\} : \quad \mathbf{y}_i^{(1)} \leq \mathbf{y}_i^{(2)} \\ \& \quad &\exists j \in \{1, \cdots, m\} : \quad \mathbf{y}_j^{(1)} < \mathbf{y}_j^{(2)}. \end{aligned} \tag{2}$$

The population is sorted in successive fronts, at which point the first Pareto front is selected to survive onto the next generation, and the last few Pareto fronts are selected to be replaced by the offspring of the surviving chromosomes generated through a crossover operation. Random mutation operations promote further evolution and optimization in the objective space.

Our application of MOO and NSGA-II follows that recommended by Khorshidi et al. (2020), with some modifications. As the motivation was not specifically to develop and train a classification model which is generalizable over unseen data sets, but rather to use the perspective gained from the optimization of ML classification models to inspect the quality of imputation carried out on a substantially empty data set, we forgo the use of a static holdout set. Furthermore, since we expect the imputation to be reliant on there being a dependence between stellar properties and planet occurrence, the imputation module is not blind to the target labels, which are provided to it as input features. Since the imputation and classification modules are done independently of one another, the Pareto fronts can still be expected to provide a balanced diverse set of solutions that are not heavily skewed by imputation models which are inaccurate and skew the imputed features to favour higher classification metrics. Furthermore, as the imputation methodology is sophisticated enough that combined with a multidimensional data set such as the Hypatia Catalog, the influence of the input features during imputation will be controlled by the input of the distributional and statistical characteristics of the chemical, physical, and galactic

Table 2. Data set completeness statistics for all feature variants. As can be seen in their completion rates, the majority of the variants are missing a substantial portion of their features, highlighting the importance of ensuring imputation is done multidimensionally to maximize the use of information present in the data set.

Data set	Num. features	Feature category			Completion
		Chem.	Phys.	Gal.	Rate (per cent)
Full (35 per cent)	39	28	7	4	64.43
Full (25 per cent)	45	34	7	4	60.18
Full (9 per cent)	51	40	7	4	54.45
Chem. only (25 per cent)	34	34	–	–	55.42
No Phys. (25 per cent)	35	34	–	4	55.07
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	27	20	7	–	72.30
Vol.+Lith.+Fe+Phys. (25 per cent)	23	16	7	–	73.99
Lith.+Sid.+Fe+Phys. (25 per cent)	25	18	7	–	72.81
All Chem.+Phys. (25 per cent)	41	34	7	–	60.96

Table 3. Data set statistics for all sample variants used in this work.

Data set	Sample size	Comparison stars (per cent)	Giant Pl. hosts (per cent)	Low-mass Pl. hosts (per cent)	Multiple Pl. hosts (per cent)
Full data set	9982	8632 (86.48 per cent)	621 (6.22 per cent)	789 (7.90 per cent)	456 (4.57 per cent)
Giant stars omitted	7244	6035 (83.31 per cent)	490 (6.76 per cent)	777 (10.73 per cent)	426 (5.88 per cent)
Low-mass stars omitted	9789	8468 (86.51 per cent)	613 (6.26 per cent)	765 (7.814 per cent)	442 (4.52 per cent)
Intermediary stars	7052	5872 (83.27 per cent)	482 (6.83 per cent)	753 (10.68 per cent)	412 (5.84 per cent)

stellar features. The level of influence of the target labels during the imputation will be a factor which needs to be controlled dependent on the use-case in which it is being applied.

2.6 Imputation of missing features

As is explained in Section 2.3, a substantial proportion of the data is incomplete. Hence, for imputation to be possible, the methodology used needs to incorporate as much dimensionality as possible for the data to be informative across all features, without introducing systematic biases by relying on extraneous and/or low-populated features. Furthermore, simplistic imputation methods such as mean imputation would not be viable in cases of high rates of missing data. This, combined with the fact that it is a validated imputation method, led Khorshidi et al. (2020) to implement fuzzy clustering in the imputation module of their recommended configuration.

Whilst a full description of fuzzy clustering is beyond the scope of this section, a brief definition will benefit in understanding the methodology of our work. Differing from the typical clustering techniques used more commonly for unsupervised ML, the fuzzy clustering algorithm proposed by Bezdek (2013) is a soft allocation clustering method, such that rather than being assigned to a single cluster, each data point is given a membership degree for all clusters. So, each data point is a member of all clusters, with the strength of their memberships varying in such a manner that they are normalized to 1. The reader is directed to section II.B of Khorshidi et al. (2020) for a mathematical description.

The manner in which fuzzy clustering is used for imputing missing values is as follows, and is unchanged from that provided by Khorshidi et al. (2020). Each chromosome in the genetic pool will encode both the number of clusters K, and the feature coordinates of cluster centres c_{ij}, where i represents the cluster number and j is the jth feature. For every chromosome, the membership of each data point is calculated as defined in equation (3) of Khorshidi et al. (2020). The membership degrees are then used to impute the missing value as a linear combination of the cluster centres. For a missing feature j of data point l, with membership degrees $\{u_{l1}, u_{l2}, \ldots, u_{lh}, \ldots, u_{lK}\}$ for cluster centres $\{c_{1j}, c_{2j}, \ldots, c_{hj}, \ldots, c_{Kj}\}$,

$$\text{Imputed Value}_{lj} = \sum_{h=1}^{K} u_{lh} c_{hj}. \tag{3}$$

It should be noted that this differs for the first generation, for which mean imputation is used to populate missing features to be able to calculate the membership degrees.

2.7 Classification models

The choice of which algorithm or model to apply for a specific ML task depends on numerous factors. It is strongly reliant on the nature and characteristics of the data set, as well as the available sample size. The computational power and time-frame for fitting/training your model will also be important to consider for practically. And of course, the task itself will have a principal role. This subsection aims to explain the motivation for our selection of supervised classification models for the set of results explained in Section 4, such that the best and most stable performer would be chosen as the model in place for the results presented in Sections 5 and 6. A general, high-level introduction to all models will also be provided for completeness.

For the initial classification model selection, we test three different models, all of which will be explained in more technical detail in Section 3. The model at this stage is tested solely as a binary classification task, to maximize the interpretability of the classification results. Once a model was chosen for the full set of results, the system was modified to accommodate for multilabel classification.

The first model implemented was the SVM algorithm, as used in the Khorshidi et al. (2020) set-up. SVMs attempt to approximate a hyperplane which maximizes the decision margin between the two classes. This is effectively done by mapping input vectors to a high-dimension feature space in which a linear decision surface is constructed (Cortes & Vapnik 1995). SVMs tend to be adept with data sets that have high dimensionality, and do not necessarily require the large sample sizes demanded by other algorithms. The latter point does however depend on the clarity of the margin of separation between the two classes. When applied to noisy data sets, it tends to suffer in performance. Furthermore, scalability is an issue, as SVMs become substantially more computationally expensive for very large data sets. None the less, the model's adaptability and interpretability makes it a good baseline classifier to include in our initial test, even more so since it is the classification model used in the original implementation.

The second model we tested was a simple dense neural network architecture. The fully connected neural network is, after the single neuron perceptron, the most fundamental architecture for artificial neural networks. Their strength lies not solely in their ability to learn complex representations in data, but in their great versatility to adapt to most ML tasks. This fluidity in design, combined with their aptitude for unearthing statistically significant representations in relatively noisy data sets provided the sample size is large enough, was the specific reason why we attempt to include in our initial tests. As will be made clear, our design was kept quite simple and our selection (and by extension its full search) of the hyperparameter space is by no means exhaustive. However, testing an initial run here will provide a springboard for future work, should the performance metrics yield promising results.

The third and final model incorporated into our MOO algorithm was the same architecture of the recommendation algorithm implemented by Hinkel et al. (2019). The XGBoost algorithm proposed by Chen & Guestrin (2016) is a powerful predictor, expanding on the standard gradient boosting framework by incorporating regularization terms in the model's objective function. Its strong performances in several ML competitions and versatility have made it a popular ML model to be applied to a different assortment of large-scale, real-world data sets. This, combined with the fact that a large aspect of this work is to comparatively assess our results with the original interpretations of the Hypatia Catalog and its applicability for training a recommendation algorithm meant that an important test, at least for the initial classifier selection, would be to include the XGBoost model.

2.8 Choice of fitness metrics

For any optimization problem, be it single or multi-objective, the choice of fitness metrics will define not solely the destination and the type of solution, but the possibility and path of the loss minimization itself. Hence, care needs to be taken on which metrics are selected.

The imputation metric we chose to implement is the first of three suggested by Khorshidi et al. (2020) in the original work. As previously mentioned and as will be explained in more detail in Section 3, our work employs the data set as one whole pool of samples available for training. Furthermore, we are particularly interested in testing the imputation module's ability to complete the data set in a way that invokes the distributional characteristics of the data across all dimensions, rather than simply from that particular feature's distribution alone. This led to the selection of a cluster validity function, which as stated in Khorshidi et al. (2020) will evaluate the performance of the clustering task, allowing for the tuning of the cluster centre positions and clustering model hyperparameters. The specific objective function used is the average silhouette width (ASW), a cluster validity index which describes the cohesion of intra-cluster data points and separation between inter-cluster data points. Reproduced from Khorshidi et al. (2020), if $a\,(l, k)$ is the average separation between data point l and other intra-cluster data points, and $b\,(l, k)$ is the minimum average separation between data point l and all inter-cluster data points, then

$$\mathrm{ASW} = \frac{1}{N}\sum_{l=1}^{N}\frac{b\,(l, k) - a\,(l, k)}{\max\{a\,(l, k)\,, b\,(l, k)\}}. \tag{4}$$

Values will be normalized between −1 and 1, and higher ASW values represent a more well-clustered data set.

The classification metrics used for optimization needed to balance prioritizing both recall of true positives (TPs) and precision in correctly classifying true negatives. Hence, both the simple and interpretable accuracy metric, and the more strict F_1 metric, were considered. As will be explained in Section 3, the large proportion of comparison stars, compared with the small sample of hosts, in both binary and multilabel variants of the data set, led to the decision to subsample the data set within the classification module. This made the baseline score more palpable in the case of the giant planet classification fitness, and thus allowed the use for the accuracy metric in the binary classification task. In the case of the multilabel modification, since the frequency of all three labels varied substantially, using accuracies no longer remained viable. Hence, F_1 scores were chosen for the multilabel for a standard, equally weighted approach. The multilabel functionality treats the score or fitness for each label as separate dimensions in the fitness space, implying that the MOO task transforms into a 4D optimization problem, rather than a 2D one. This allows us to explore the trade-off for each label in more detail, to ensure that the selected Pareto fronts prioritize performance over all four fitnesses equally.

It should be highlighted that for the classification fitness/es to validate the model's performance, they still need to be performed over a holdout set, either in the form of cross-validation techniques or a static holdout set. Again, this will be further explained in Section 3.

As a final point, it is valuable to stress that whilst the classification fitnesses are indeed primarily a measure of performance by the classifier, they do provide valuable insight on the quality of imputation. Whilst it is possible that incorrect imputation can lead to strong performances, simply by separating the individual classes more than it should by maximizing and overexpanding their discriminating features, a diverse set of Pareto non-dominated solutions will explore the trade-off between all fitness functions to find the best combination for all.

3. SYSTEM DESIGN

Once the methodology was devised, the next phase was its implementation. To ensure the recommendation in Khorshidi et al. (2020) are followed, the MOO implementation was done manually, in which we define not only imputation and classification modules but also tournament selection and the crossover and mutation operators. The remainder of this section will explain the major design choices within these modules. The reader should be reminded that the system design is based on the approach given by Khorshidi et al., but deviates in implementations of particular choices in the modules, especially in the case of our modification for multilabel classification and 4D optimization. We will fully explain our implementation, but will also suggest to the reader to consult Khorshidi et al. (2020) where an unchanged component is not extensively described.

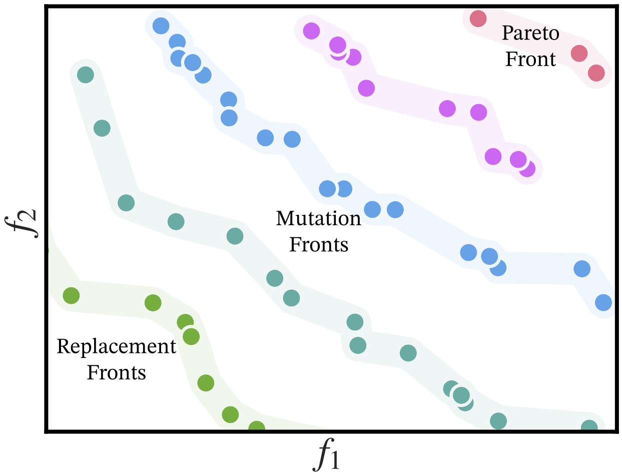

Figure 1. Schematic representation of the non-dominated sorting in the fitness space, such that all chromosomes belong to a set within which they do not dominate. The lowest performing fronts are selected for replacement from offspring chromosomes, and the best performers, the first Pareto front, survive unscathed into the next generation. The fronts in between are subject to mutation operations prior to being transferred into the next generation to promote genetic diversity.

3.1 NSGA-II MOO implementation

The design of the NSGA-II algorithm closely follows that described by Khorshidi et al. (2020). An initial chromosome population of size 100, each encoding a particular hyperparameter tuning for both imputation and classification modules, is randomly generated. For every chromosome, the missing features in the data set are imputed, as described in Section 2.6, using mean imputation for the first generation and then fuzzy clustering membership imputation for subsequent generations. With the data set complete for that particular chromosome, the imputation and classification metrics are calculated, as will be explained in Sections 3.5 and 3.6.

Once the fitnesses of all chromosomes have been determined, tournament selection can occur, in which the algorithm sorts the population into successive fronts, disjoint subsets of the population within which the chromosomes are non-dominating. The method in which this is done follows the mathematical description given in Emmerich & Deutz (2018) and in equation (2), and through the use of NUMPY functionalities achieves sufficiently fast computation. A schematic representation of the result of this mechanism is found in Fig. 1.

The fronts will then determine which chromosomes survive onto the next generation, are candidates to be selected to parent new offspring chromosomes, and are likely to undergo mutation. The first Pareto front, meaning the set of chromosomes which are non-dominating with respect to each other yet dominate the rest of population, will survive onto the next generation unscathed, immune even to mutation operations through the transition. On the other end of the performance spectrum, a predefined number of the lowest fitness fronts, which are dominated by the rest of the fronts, are chosen to be killed off and replaced by offspring chromosomes generated by the crossover operator. The number of fronts replaced by new offspring was selected in such a way that the first couple of generations saw around 10–20 per cent of the population fall within these replacement fronts, solely to ensure that the explored hyperparameter space is deemed to initially broad enough for as full an exploration as possible. After a small initial exploration during the testing stage, we determined that this was reliably achieved with a configuration setting of three replacement fronts for the binary classification runs and setting it to four replacements fronts for the multilabel modification. Finally, those surviving chromosomes that fall within a front in between the first Pareto front and the low-performing replacement fronts, are all subject to mutations governed by the mutation operator described below.

3.2 Chromosome design

A properly designed chromosome in a GA needs to provide a unique encoding/representation/description of the model, such that the resulting fitness/es can result in stable minimization of the objective function. As explained in Khorshidi et al. (2020), for the type of MOO task this work attempts to tackle, in which both two independent modules require hyperparameters settings, the chromosome needs to contain all settings for both. Hence, the chromosome can be seen to be split into two particular sections: the imputation genes, which control to fuzzy clustering parameters which set the model used to impute values for missing features, and the classification genes, which are the selected hyperparameter values for the particular ML algorithm/network applied. This is schematically represented in Fig. 2, which is adapted from fig. 3 in Khorshidi et al. (2020), and modified to suit our specific usage of the design. The specific genes selected within each section will be explained further in Sections 3.5 and 3.6.

3.3 Crossover operator

The importance of a well-defined crossover operator, the component of the evolutionary algorithm tasked with generating the offspring chromosomes to be introduced into the population for the following generation, cannot be understated. Its role is to govern the exploration of the parameter space in such a way that the general fitness of the population improves over successive generations whilst at the same time maintaining genetic diversity within the sample pool. In our work, we adopt the operator described in Khorshidi et al. (2020), so this subsection will only provide a brief description to provide context. The reader is directed to the original work for a full description on the motivations behind the design.

At the point when the operator is invoked, the number of chromosomes N_R to be replaced has already been identified. As will be described shortly, each usage of the crossover operation will generate a total of three offspring. Hence, the operation itself will be repeated several times until N_R offspring chromosomes have been generated.

Focusing on the mechanism at play within each crossover operation, the first step is to subsample the population (including the replacement fronts, to ensure chromosome diversity is preserved) to select a number of parental candidates C_R. In all runs leading to the results presented in this work, this number was chosen to be 30 per cent so as to present a substantial portion of the total population whilst at the same time being as small enough percentage to promote greater variation in the parental pool amongst all generated offspring chromosomes. Out of this parental pool, three pairs of parents are selected (leading to a single offspring each): the best performer of the imputation objective function paired with the best performer of the classification objective function; the best imputation performer paired with the worst classification performer; and the worst imputation performer paired with the best classification performer. The latter two are included for genetic diversity. In the case of the multilabel pipeline, to preserve the ordering applied in the work and respect the genetic diversification introduced here, we

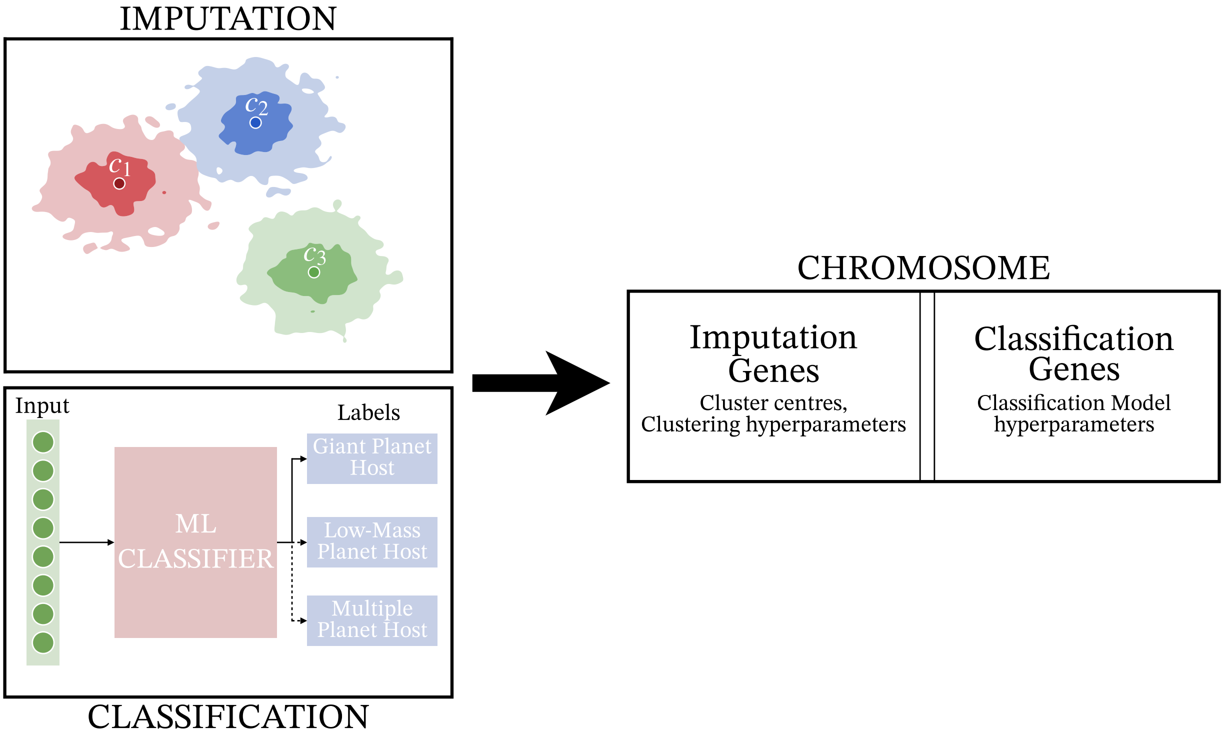

Figure 2. Schematic for the chromosome encoding within the GA design. The specific configuration for both the imputation and classification model is represented as a string of genes within the chromosome. The set of imputation genes consists of the clustering hyperparameters and the coordinates of all designated cluster centres. The classification genes will be used to build and define the classification model, and therefore depend on whichever model is being utilized in that particular run. Within the classification module, schematic solid directional lines represent paths which are present in both the binary and multilabel variations of the design, while the dashed line is used to represent those present only in the multilabel modification.

decided to replace the single classification objective function score with a mean of the three classification objective functions.

Once the parent-pairs are chosen, the offspring is generated as follows. First, the size of the offspring is randomly chosen as one of the sizes of its parents. Then, a binary vector of the same length as the 'shorter' parent chromosome is randomly generated, which will determine from which progenitor the corresponding gene will be taken from. In most cases, where the two parental chromosomes are of unequal length due to the different number of cluster centres, those centres which do not line up with the binary vector will be appended and directly transferred to the offspring from the 'longer' parent. An important factor one can notice from this mechanism, one which is not mentioned in the description in Khorshidi et al. (2020), is that this method can potentially lead to degeneracies of similarly positioned cluster centres. However, it should not be expected to affect the imputation of missing features if two cluster centres are degenerate, and thus can be handled during the post-run analysis.

3.4 Mutation operator

The simplest to implement out of all the GA components the mutation operation will promote genetic diversity and further exploration of the parameter space by randomly altering a number of genes within the chromosome. The mechanism works as follows. The number of genes to be mutated is randomly generated through a binary vector in which 1 implies that a mutation will occur to the corresponding gene. Then, every gene is mutated by randomly generating a new parameter setting within a specified range of possible values, which depending on the gene can be a continuous or discrete set.

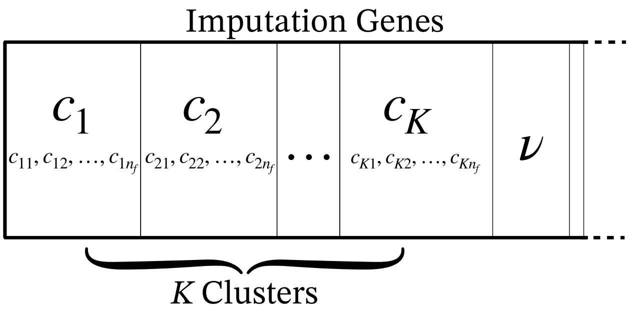
Imputation Genes
c_1
$c_{11}, c_{12}, \ldots, c_{1n_f}$
c_2
$c_{21}, c_{22}, \ldots, c_{2n_f}$
$\cdots$
c_K
$c_{K1}, c_{K2}, \ldots, c_{Kn_f}$
ν
K Clusters

Figure 3. Design of imputation gene within the chromosome. For the initial population generation, a 'proto'-chromosome is generated with the assigned values of K and ν.

3.5 Imputation module

For the module in the pipeline responsible for using C-Means fuzzy clustering to impute all missing features across every sample in the data set we directly apply the system suggested by Khorshidi et al. (2020), so a more complete description can be found in their work.

In essence, the chromosome will encode all necessary model parameters describing the clustering model, such that the membership degrees can be calculated and the imputed values are updated. Within the chromosome's imputation representation, a model with a number of clusters K and fuzziness parameter ν, applied to a data set of sample size n_s and number of feature n_f is encoded as follows: The first $n_f K$ genes represent the coordinates for all cluster centres of the model, hence specifying both the values of K and cluster centre positions, and then followed by ν, as shown as Fig. 3. Therefore as

Table 4. The accepted ranges for imputation genes within the 'proto'-chromosome. The full chromosomes are then generated using SciKit Fuzzy's C-Means fuzzy clustering.

Imputation genes	
Gene	Range
Number of clusters (K)	2–15
Fuzziness (ν)	1.5–5.0

the number of clusters is a free variable, this introduces a variation in the size of the chromosomes amongst the population. This, as is the case in the work of Khorshidi et al. (2020), is accounted for in the crossover operator and fully described in Section 3.3.

For the first generation, the population is initialized without the cluster centre coordinates. For these 'proto'-chromosomes, the values K and ν are randomly generated within their respective accepted ranges, given in Table 4. After mean imputation is used to complete the data set, standard normalization is applied and for every the proto-chromosome, the `cmeans` function from the SciKit Fuzzy[7] `skfuzzy.cluster` clustering submodule is used to determine the cluster centre coordinates for that particular proto-chromosome configuration. Finally to ensure genetic diversity and minimize immediate degeneracies, prior to completing the full genetic make-up, we incorporate a uniformly distributed jitter term to the cluster centres, with its centre at zero and maximum possible size determined by the respective feature's standard deviation in the data set prior to imputation. Once calculated, the 'full'-chromosomes are prepared and calculation of the fitness functions can occur.

3.6 Classification module

As is the case in any typical ML task, the classification pipeline includes the final processing of the now-imputed data set, building the classification model as per the specifications in the chromosome, and finally incorporating a training and testing regime to provide classification metrics to gauge the model's ability to approximate a generalizable transformation function. This subsection will explain the pipeline in detail, followed by the configurations of all three models tested in the initial classifier selection set of runs.

An initial exploration of the data set, as explained in Section 2 and demonstrated by the statistics in Section 3, immediately shows that all hosts within the full sample only constitute a small percentage, ranging from 13.49 to 16.73 per cent across all sample variants. Dealing with heavily imbalanced data sets, whilst common in real-world domains, is an important factor for which a strategy needs to be put in place for an effective analysis of model performance and generalization (Kotsiantis et al. 2006). As the comparison sample is relatively large and diverse, we elect to employ random undersampling of the comparison set, such that the host sample remains entirely used and constitutes a larger portion of the final data set used for training/fitting and testing. To further ensure stability in performance, for every chromosome the data set undersampling and subsequent training/fitting is repeated four-times. The proportion of which the comparison stars constitute the final subsample is also a configurable setting in our pipeline. For the initial classifier selection runs presented in Section 4, the setting was set to 0.6, meaning that 40 per cent of the subsample on which the model is trained and tested are giant planet hosts. For the results in Sections 5 and 6, since the data set now has multiple labels contend with, and so that both sets of results can be cross-examined, the setting was higher for giant planets in the full data set, with a value of 0.816. This was due to the fact that in this case, the *entire* host sample constitutes 40 per cent. For low-mass planet hosts and multiplicity, the setting would equate to 0.766 and 0.865. This implies that the baseline accuracies expected for the three labels are 81.6 per cent, 76.6 per cent, and 86.5 per cent, respectively.

Table 5. The accepted ranges for classifications genes within the chromosome, in the case of when the SVM model was implemented.

SVM classifier genes	
Gene	Range
Flexibility (C)	[10^{-3}, 100]
Kernel function (K_r)	[1, 2, 3, 4]
γ	[5×10^{-4}, 5]
d	[2, 7]
r	[0, 40]

To ensure that the performance metrics constrain the model ability to generalize and predict unseen data, we employ k-fold cross-validation. The subsample was further sampled in a stratified strategy into five folds, and the model was validated on each after being trained on the other four. Hence, combined with the fact that this entire procedure was repeated for four different subsamples, all performance metrics were obtained a total of 20-times, which allowed for the opportunity of an associated error with our classification scores, which were taken as the median of the score sample.

Moving onto the classifier design, as mentioned in Section 2.7, we implement three different models in the initial classifier selection tests (Section 4), to then use the best (and most stable) performer for our analysis in the subsequent sections. It should be noted at this stage that prior to the use of any of the following models, the data set was passed through a standard scaler.

The first model, based off of reasons mentioned in Section 2.7 and due to it being the model recommended by Khorshidi et al. (2020), was an SVM model, implemented using Scikit-Learn's `svm`[8] module's SVC function. We follow the implementation in the original work, so the following description can be found in more detail there. Five hyperparameters are set as free variables within the optimization task. First, the flexibility regularization parameter C is set. The remaining four all relate to the kernel function being implemented. The parameter K_r set to one of four options for the function chosen, defined as follows for data points x_i and x_j,

$$\begin{aligned}
\text{Linear Kernel } (K_r = 1), &\quad k\left(x_i, x_j\right) = x_i \cdot x_j \\
\text{Radial Kernel } (K_r = 2), &\quad k\left(x_i, x_j\right) = \exp\left(-\gamma \parallel x_i - x_j \parallel^2\right) \\
\text{Polynomial Kernel } (K_r = 3), &\quad k\left(x_i, x_j\right) = \left(\gamma\left(x_i \cdot x_j\right) + r\right)^d \\
\text{Sigmoid Kernel } (K_r = 4), &\quad k\left(x_i, x_j\right) = \tanh\left(\gamma\left(x_i^T \cdot x_j\right) + r\right)
\end{aligned} \tag{5}$$

where γ, d, and r are the three remaining free hyperparameters. The accepted ranges for all are presented in Table 5, and were set to ranges based on those chosen by Khorshidi et al., slightly widened to attempt to explore the parameter space further.

The second model implemented was a dense Artificial Neural Network (ANN) architecture which is both adaptable enough to offer diversity across the chromosome pool whilst simultaneously design in a way so as to expect strong performance. Keras Sequential

[7] https://pythonhosted.org/scikit-fuzzy/

[8] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.svm

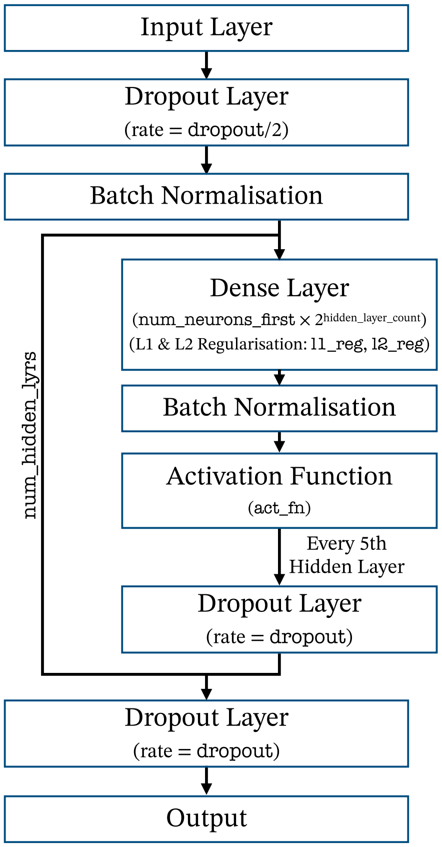

Figure 4. Model design for the Dense ANN classifier.

API was used to build the model depending on a total of eight free hyperparameters, double the amount in the SVM model. Whilst this immediately increases the complexity of the optimization task, the fact that a number of these hyperparameters are regularization parameters, then it is increasingly likely that the set of final Pareto fronts of model parameters should be ones which not only maximize performance, but also minimize the risk of overfitting due to the statistical measure incorporated into the final fitness metric.

As can be seen in Fig. 4, the input layer is immediately followed by two regularization layers. First a dropout layer set to have a dropout rate equal to *half* the dropout value used later in the network (set by the hyperparameter `dropout`, to ensure an informational bottleneck is not instigated in cases where a higher dropout rates are tested deeper in the network. This is then followed by a batch normalization layer, set to the default settings provided by Keras. Then, depending on the number of hidden layers determined by the value of `num_hidden_lyrs` iteratively adds dense layers followed by a batch normalization layer and with the activation function placed *after*, as suggested by Ioffe & Szegedy (2015). The number of neurons in the first hidden layer is set by `num_neurons_first`, a number which doubles with each successive hidden layer. The dense layer also has both L1 and L2 kernel regularization implemented within it, the strength of which is determined by the hyperparameters `L1_reg` and `L2_reg`, respectively. The type of activation function used for every hidden layer is determined by another free hyperparameter, the value of which is set by `act_fn`. It should be noted that out of concern of low regularization in deep networks, every fifth hidden layer (unless it is the final hidden layer) is followed by dropout layer set to a dropout rate equal to `dropout`. Finally, once all hidden layers have been added, we incorporate one final dropout layer set to a dropout rate of `dropout`, and follow it with an output layer of one neuron (for binary classification; this would be switched to three neurons if chosen for multilabel classification) with a sigmoid activation function. Once fully built, the model is compiled with a `binary_crossentropy` loss function defined as

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \cdot \log\left(p\left(1 - y_i\right)\right) \quad (6)$$

and optimizer defined by the hyperparameter `optimizer`. The final hyperparameter `num_epochs`, sets the number of epochs for training, and is therefore external to the model, rather than pertaining to the architecture itself. The list of accepted values for all eight hyperparameters is provided in Table 6.

The third and final classification algorithm tested in this work is the XGBoost model, as was applied in the analysis by Hinkel et al. (2019). The model was implemented using the Python API for XGBoost,[9] an optimized distributed gradient boosting library. For multilabel classification, Keras `MultiOutputClassifier` function from its `multioutput` module was used to implement a multilabel strategy with the XGBoost model fed into it after it was initialized. Focusing on said initialization, after the objective was set to logistic binary, eight free hyperparameters were selected for optimization through the GA, varying from being architectural to being fully regularization parameters: `n_estimators`, which sets the number of boosting rounds and represents the total number of trees in the forest. `max_depth` will determine the maximum depth of a tree, avoiding overfitting by limiting the number of nodes in the tree. The `learning_rate` determines the step size shrinkage used to prevent overfitting, scaling the contribution of each tree and therefore affecting optimization. The `gamma` parameter determines the minimum loss reduction necessary to create a further partition on a leaf node. It helps in controlling the complexity of the tree by specifying the minimum reduction in the loss function required to make a split. Sampling strategies to incorporate further regularization within the model can be set by the `subsample` and `colsample_bytree` parameters, which set the fraction of training data and features, respectively, to be randomly sampled for each boosting round. Both allow for the model to attempt to learn representations across samples and features which would aid in stronger generalization. The final two hyperparameters are purely regularization terms to the loss function, as was applied in the dense ANN model with the kernel regularizers. L1 regularization is set with `reg_alpha`, whilst L2 penalties are set with `reg_lambda`. The selected ranges for all classifier genes for the XGBoost model are given in Table 7.

The final classification genes for all three types of runs are shown in Fig. 5.

4. INITIAL CLASSIFIER SELECTION

Once all three classification configurations were built within their respective MOO implementations, the initial set of exploratory runs was launched. The aim for this set of results was to select which classification model demonstrated the strongest and most stable performance. Hence, this section will focus on the general performance of the MOO algorithm and the classification module in terms of the chromosome population, rather than focus on the individual chromosomes and their model's representation of the data set.

The first interpretable result from the MOO runs is the evaluation of the position of the final Pareto front of the chromosome population in

[9] https://xgboost.readthedocs.io/en/stable/python/python_api.html

Table 6. The accepted ranges for classifications genes within the chromosome, in the case of when the dense ANN model was implemented.

Dense ANN classifier genes			
Gene	Range	Gene	Range
`dropout`	[0, 0.5]	`act_fn`	[‘`selu`’, ‘`elu`’, ‘`relu`’, ‘`tanh`’]
`num_neurons_first`	[5, 30]	`num_hidden_lyrs`	[2, 10]
`L1_reg`	0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1	`L2_reg`	0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1
`optimizer`	[‘`adam`’, ‘`adagrad`’, ‘`nadam`’, ‘`sgd`’]	`num_epochs`	[50, 250]

Table 7. The accepted ranges for classifications genes within the chromosome, in the case of when the XGBoost model was implemented.

XGBoost classifier genes			
Gene	Range	Gene	Range
`n_estimators`	[50, 500]	`max_depth`	[5, 50]
`learning_rate`	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1	`gamma`	[0, 10]
`reg_lambda`	0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1	`reg_alpha`	0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1
`subsample`	[0.5, 1]	`colsample_bytree`	[0.5, 1]

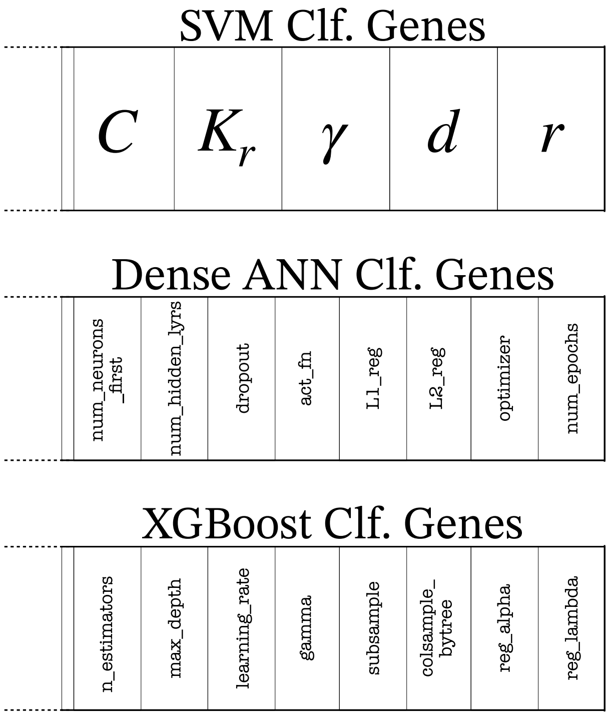

Figure 5. Design of classification genes within the chromosome for all three model implementations.

the fitness space. This not only allows for an immediate comparative assessment of the performance of the three models, but additionally provides context on their effect on the trade-off between both fitness functions in the binary classification runs. As can be seen in Fig. 6, a clear trade-off curve appears for all three models, with varying degrees of deterioration at extreme values. Focusing first on classification fitness, as expected, the SVM model is the overall weakest performer. Whilst it does exhibit clear signs of generalization by outperforming the baseline expected from a random classifier, the maximum accuracy reached is substantially less than the other two models. The XGBoost on the other hand manages to marginally outperform the ANN model, securing a Pareto front which is substantially more stable in its strong performance even for significantly different degrees of imputation fitness. The ANN model, whilst able to attain relatively strong generalization, suffers a significant degradation in its trade-off at higher imputation fitnesses. Indeed, its maximum imputation fitness is significantly lower than the other two models, suggesting that the current network architecture tends to perform better with data which is not as clearly clustered. As one particular aim of this work is to investigate the clustering implemented in the imputation module, this incompatibility with well-defined clusters was deemed to potentially hinder subsequent analysis.

To complement the graphical interpretation, we calculate the area under curve (AUC) of all three Pareto fronts to quantitatively represent the front’s maximization of the trade-off between both fitness functions, with the ideal value being an elbow at (1.0, 1.0). The boundaries of the imputation and classification fitnesses were taken to be $[-1, 1]$ and [0, 1], respectively, using this to normalize the scores to be within [0, 1]. The values can be found in the legend in Fig. 6, and clearly show that the XGBoost model is the clear best performer. The ANN marginally outperforms the SVM, mainly due to its overall greater margin of performance in classification fitness than its underperformance in imputation. Table 8 mirrors this, demonstrating that the XGBoost classifier achieves higher scores in both fitness functions, which when considered in conjunction to its curve in Fig. 6, does so without suffering a severe drop in classification scores at higher imputation values.

As this section mainly involves the investigation of the classification module’s performance within the overall context of the MOO algorithm, we choose to explore the classification performance at two regions of the fitness space, one at low imputation fitness and one at high values. Selection of which pair of chromosomes to select for each model turns out to be relatively simple. The trade-off curves show that the low imputation fitness region is achieved by the chromosomes which tend to score the highest classification fitness, and vice-versa for the high imputation fitness region. The sets of results in Table 9 hence respectively choose the best and worst classification performers for each model. As these sets of results were re-generated after the MOO implementation to be able to generate all necessary performance metrics and plots, it is important to note that values for the median score will not precisely agree with those visible in Fig. 6,

Table 8. General overview of performance metrics of the Pareto fronts of all three models. As can be seen, the XGBoost outperforms both the SVM and ANN in both fitness functions.

Clf. model	Pareto front size	Max. Imp. fitness	Max. Clf. fitness
SVM	9	0.8356	0.8580
ANN	8	0.6557	0.9418
XGB	7	0.8598	0.9578

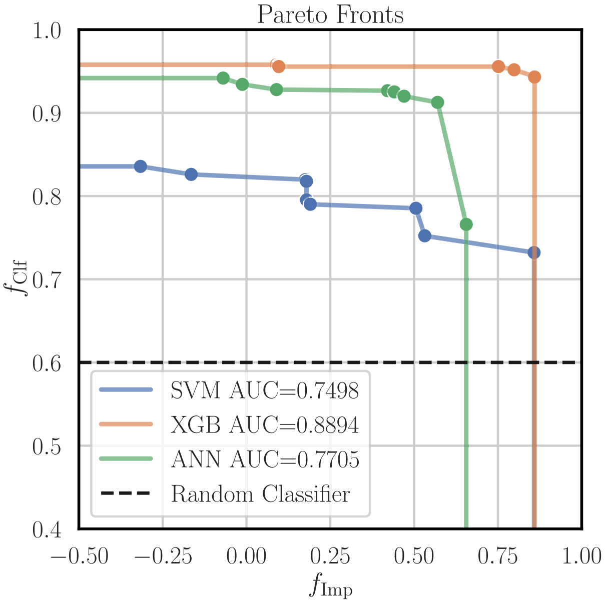

Figure 6. Final Pareto fronts for the initial runs to allow for selection of the classification module to be implemented. The XGBoost classifier significantly outperforms the other two, with the ANN model also improves on the SVM in the classification fitness. However, the ANN suffers from a sharper deterioration in classification fitness than the SVM when imputation fitness improves. The dashed line represents the baseline score which would be achieved by a random classifier. We also present normalized AUC scores for all three Pareto fronts.

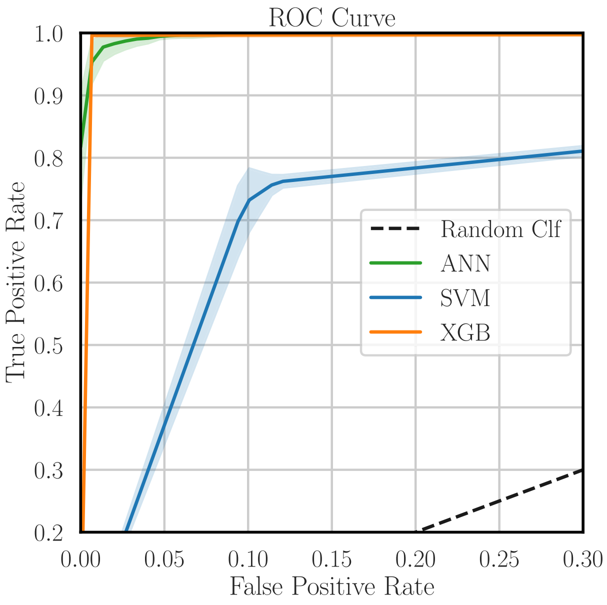

Figure 7. ROC curves for the highest classification performers for each model in their respective final Pareto front.

but all fall within the associated error. Both sets of results corroborate the performance shown in the fitness space, with the XGBoost model displaying a significantly more stable performance across its chosen set of chromosomes. Its stability further extends across the subsampling and cross-validation strategies, demonstrating a lower error for most performance metrics (all bar a marginally larger error than the ANN model in the recall of the worst Pareto classifier). The receiver operating characteristic (ROC) curves are shown in Fig. 7 for the best performers and in Fig. 8 for the worst performer. The AUC scores of these ROC curves are given in Table 9 and provide further evidence for the greater consistency of the XGBoost. Whilst the ANN model shows strong performance for its best-classifying chromosome which is comparable with that of the XGBoost, the ROC curve has a far larger uncertainty band suggesting less stability. Indeed, in the worst-classifying chromosome's performance, the deterioration is drastic, with the band growing even larger. The XGBoost however maintains its tight uncertainty band across the change in chromosome, scoring a high AUC in both.

Hence upon reviewing all the results, it was clear that the best and most consistent performer was the XGBoost model. This means that for the remainder of the results, we employ this model into the classification module so as to focus on exploring both the feature space, in terms of the distributional characteristics uncovered and the selection of input features to feed to the classifier, and the sample space to explore dependencies on which families of stars are included in the analysis.

Table 9. Performance metrics of the set of highest and lowest classification performers for each model in their respective final Pareto front. As these sets of results were re-generated after the MOO implementation to be able to generate all necessary performance metrics and plots, it is important to note that values for the median score will not precisely agree with those visible in Table 8, but all fall within the associated error.

			Best classification performers		
Clf. model	Accuracy	Precision	Recall	F_1 score	ROC-AUC
SVM	0.818 ± 0.021	0.739 ± 0.031	0.848 ± 0.035	0.789 ± 0.022	0.823 ± 0.020
ANN	0.918 ± 0.036	0.890 ± 0.041	0.908 ± 0.066	0.898 ± 0.048	0.916 ± 0.041
XGB	0.951 ± 0.011	0.938 ± 0.019	0.941 ± 0.019	0.939 ± 0.014	0.949 ± 0.012
			Worst classification performers		
Clf. model	Accuracy	Precision	Recall	F_1 score	ROC-AUC
SVM	0.724 ± 0.026	0.644 ± 0.038	0.701 ± 0.024	0.671 ± 0.025	0.720 ± 0.024
ANN	0.737 ± 0.024	0.699 ± 0.039	0.606 ± 0.041	0.648 ± 0.033	0.715 ± 0.025
XGB	0.940 ± 0.014	0.936 ± 0.022	0.914 ± 0.029	0.924 ± 0.018	0.936 ± 0.016

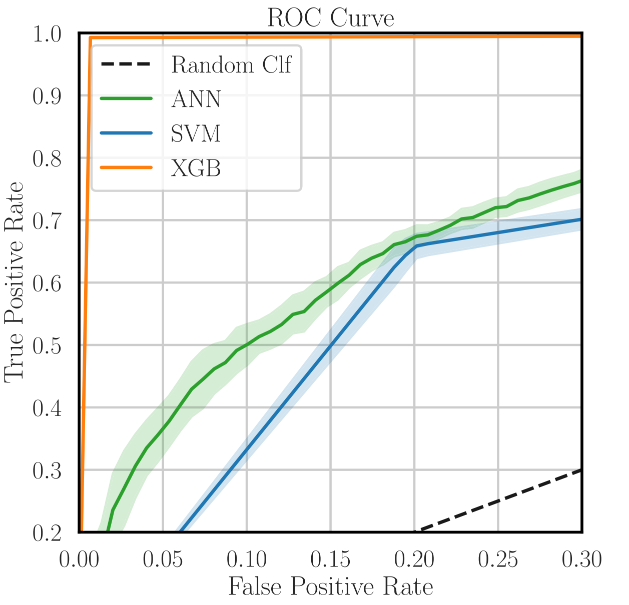

Figure 8. ROC curves for the lowest classification performers for each model in their respective final Pareto front.

5. BINARY CLASSIFICATION

With the XGBoost selected as the classifier to be implemented in the MOO algorithm, the main results of this work could finally be generated and assessed. This section will investigate the performance of the system across the feature and sample variations of the data set. The first part for each of these two sets of results will collectively describe the Pareto fronts and final chromosome selection. Then, for the feature variants, the selected chromosomes will be used to investigate the performance of the imputation module and attempt to constrain its dependency on feature and data set completion rates, and sample size. The focus is placed on feature variants because the analysis would benefit from a variant set with contrasting data set completion rates. Furthermore, since the constraints will aim to inform future feature selection vis-a-vis sample size, rather than directly affecting sample selection, it would be more prudent to work with the former set.

5.1 Feature variation

5.1.1 Pareto fronts and classification

A comparative look at the Pareto fronts and the classification metrics for a shortlist of chromosomes within the front highlights certain trends within the model's ability to impute and learn, dependent on the features present across the data sets. At this stage, whilst a look at the mean performance of the entire Pareto front for that particular run may perhaps provide a more generalized interpretation, it may potentially lack the context provided by an evaluation of different points across the front. Furthermore, a general understanding may none the less be attained by investigating over the *shape* of the front, through the use of AUC metrics and a graphical interpretation.

Upon inspection of the Pareto fronts plotted in Fig. 9, it is challenging to discern any clear dependencies on the specific absence of any particular features. Certain runs, specifically the *Vol.+Lith.+Sid.+Fe+Phys.* variant and, to a lesser extent, *Lith.+Sid.+Fe+Phys.*, lack the front sample size needed to definitively describe their Pareto fronts, and therefore claims regarding their AUC score need to be conservative. The *full data set with a 25 per cent threshold* tends to show the least forgiving trade-off between classification and imputation fitnesses, which contrasts to the performance seen for the *full data set with 9 per cent threshold*. This behaviour, combined with the performance of the *Chem. only* data set and consideration of their completion rates, confirms a suspicion of the completion rate having a significant impact on overall performance. This will be evident further on upon evaluation of the classification metrics. The more incomplete the data set, the greater the influence of the imputation module, and the greater the injection of a signal provided by the label feature at that stage. This inherently may not be a negative, and depends entirely on the context in which this MOO model is generally used. To evaluate its influence, we plot the AUC score metric against completion rate in Fig. 10, to observe whether or not both are correlated albeit noting that the sample size is limited. We find a Pearson correlation coefficient of $r = -0.12449$, implying that there is a small, but non-negligible anti-correlation. Hence, in terms of the overall resultant shape of the Pareto front, the completion rate will have a minor impact in decreasing the overall spike in performance attributed to the injected signal through imputation.

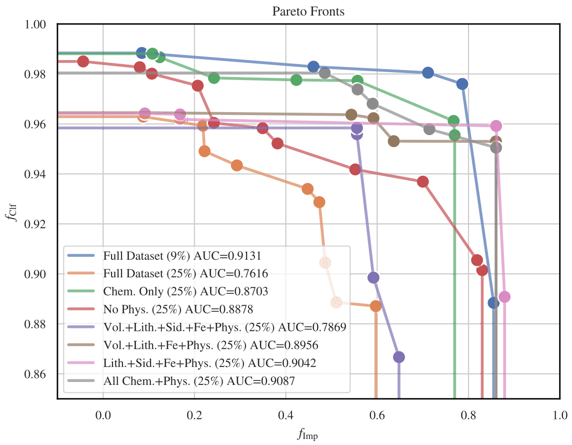

Figure 9. The Pareto front for all feature variants, with the corresponding AUC score provided in the legend.

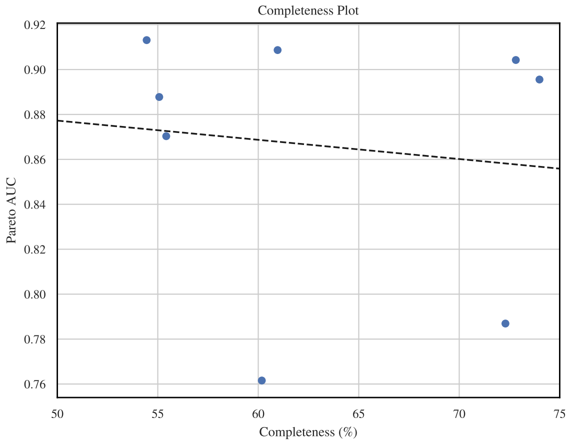

Figure 10. A plot of the AUC score against the completion rate for all feature variants. With a Pearson correlation coefficient of $r = -0.12449$, there is a small, but non-negligible anti-correlation.

Shifting focus to the classification metrics in Table 10, the discussion benefits from a comparative look at single chromosomes

Table 10. Performance metrics of the set of highest, lowest, and median classification performers for each data set feature variant in their respective final Pareto front for the binary label implementation.

		Best classification performers				
Data set	Imputation fitness	Accuracy	Precision	Recall	F_1 score	ROC-AUC
Full data set (25 per cent)	0.0877	0.9629 ± 0.0115	0.947 ± 0.0224	0.8455 ± 0.0494	0.8929 ± 0.0347	0.9174 ± 0.026
Full data set (9 per cent)	0.0848	0.9884 ± 0.0037	0.9706 ± 0.013	0.9662 ± 0.0126	0.9683 ± 0.0099	0.9798 ± 0.0067
Chem. only	0.1081	0.9881 ± 0.004	0.9765 ± 0.0152	0.959 ± 0.0138	0.9675 ± 0.0107	0.9768 ± 0.0073
No Phys. (25 per cent)	−0.0437	0.9850 ± 0.0042	0.9756 ± 0.0138	0.9421 ± 0.018	0.9584 ± 0.0118	0.9683 ± 0.0092
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	0.5558	0.9584 ± 0.0069	0.9216 ± 0.0232	0.8471 ± 0.0387	0.8821 ± 0.0213	0.9153 ± 0.0185
Vol.+Lith.+Fe+Phys. (25 per cent)	0.0904	0.9645 ± 0.0066	0.9425 ± 0.0248	0.8612 ± 0.0457	0.8988 ± 0.0213	0.9245 ± 0.0211
Lith.+Sid.+Fe+Phys. (25 per cent)	0.0914	0.9642 ± 0.0072	0.9456 ± 0.0213	0.8556 ± 0.0414	0.8976 ± 0.0225	0.9222 ± 0.0199
All Chem.+Phys. (25 per cent)	0.4852	0.9804 ± 0.0054	0.9655 ± 0.0222	0.9272 ± 0.0278	0.9455 ± 0.015	0.9598 ± 0.0132
		Worst classification performers				
Data set	Imputation fitness	Accuracy	Precision	Recall	F_1 score	ROC-AUC
Full data set (25 per cent)	0.5971	0.8871 ± 0.0123	0.8063 ± 0.0372	0.5085 ± 0.0647	0.6217 ± 0.0538	0.7405 ± 0.0321
Full data set (9 per cent)	0.8550	0.8884 ± 0.0111	0.8025 ± 0.0457	0.5234 ± 0.0426	0.6327 ± 0.0396	0.7471 ± 0.0223
Chem. only	0.7697	0.9555 ± 0.0069	0.9522 ± 0.0148	0.7988 ± 0.035	0.8683 ± 0.0223	0.8949 ± 0.0175
No Phys. (25 per cent)	0.8300	0.9015 ± 0.0107	0.8283 ± 0.0394	0.5878 ± 0.0474	0.6865 ± 0.0382	0.7801 ± 0.024
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	0.6477	0.8667 ± 0.0118	0.7679 ± 0.0542	0.3955 ± 0.0562	0.5202 ± 0.0556	0.6842 ± 0.0281
Vol.+Lith.+Fe+Phys. (25 per cent)	0.8603	0.9501 ± 0.0093	0.9079 ± 0.0275	0.8117 ± 0.0422	0.8564 ± 0.028	0.8965 ± 0.0213
Lith.+Sid.+Fe+Phys. (25 per cent)	0.8790	0.8908 ± 0.0094	0.8004 ± 0.0375	0.5427 ± 0.0383	0.6461 ± 0.034	0.756 ± 0.0196
All Chem.+Phys. (25 per cent)	0.8603	0.9505 ± 0.0089	0.8887 ± 0.0302	0.8371 ± 0.0386	0.8614 ± 0.0259	0.9066 ± 0.0193
		Median classification performers				
Data set	Imputation fitness	Accuracy	Precision	Recall	F_1 score	ROC-AUC
Full data set (25 per cent)	0.4477	0.934 ± 0.0083	0.8952 ± 0.0367	0.7279 ± 0.0329	0.8022 ± 0.0251	0.8542 ± 0.0165
Full data set (9 per cent)	0.7113	0.9805 ± 0.0044	0.9744 ± 0.0148	0.9187 ± 0.0263	0.9454 ± 0.0128	0.9566 ± 0.0125
Chem. only	0.4231	0.9776 ± 0.0032	0.959 ± 0.0198	0.9183 ± 0.0158	0.9379 ± 0.0087	0.9547 ± 0.0069
No Phys. (25 per cent)	0.3492	0.9583 ± 0.0075	0.935 ± 0.0225	0.8318 ± 0.0376	0.8798 ± 0.023	0.9093 ± 0.0186
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	0.5558	0.9559 ± 0.0071	0.9236 ± 0.0249	0.8298 ± 0.0316	0.8737 ± 0.0211	0.9071 ± 0.0158
Vol.+Lith.+Fe+Phys. (25 per cent)	0.5919	0.9623 ± 0.0064	0.9339 ± 0.0241	0.8572 ± 0.0446	0.8928 ± 0.0204	0.9216 ± 0.0206
Lith.+Sid.+Fe+Phys. (25 per cent)	0.1684	0.9621 ± 0.0081	0.9259 ± 0.0214	0.864 ± 0.0412	0.8933 ± 0.0244	0.9242 ± 0.0204
All Chem.+Phys. (25 per cent)	0.5900	0.9681 ± 0.0082	0.9456 ± 0.0201	0.8781 ± 0.0461	0.9098 ± 0.0251	0.9333 ± 0.0224

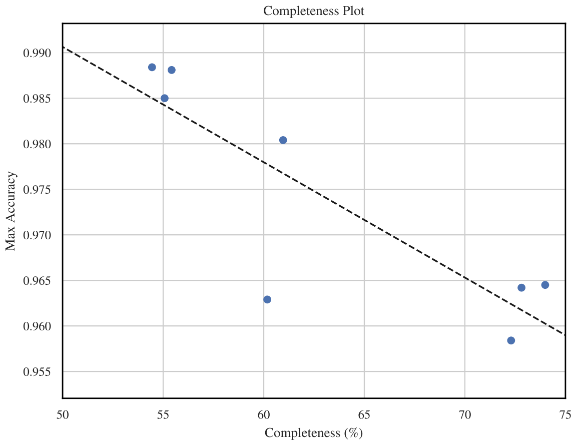

Figure 11. A plot of the maximum accuracy against the completion rate for all feature variants. With a Pearson correlation coefficient of $r = -0.85332$ there is a strong anti-correlation.

across the Pareto fronts, namely the best and worst classifiers, and the median performer. It becomes immediately evident that the signal-injection phenomenon hinted at in the Pareto front plots is more pronounced here, with data sets having the lowest completeness rates, namely the *full data set with 9 per cent threshold*, and the *Chem. only* and *No Phys.* demonstrating the strongest scores for the best classification performers. The heights reached by the classification module (albeit at the expense of the imputation ASW fitness) seem to depend on the level of influence of the imputation. We illustrate this with a plot in Fig. 11 of the max accuracy against the completion rate, and the strong anti-correlation is immediately evident. The Pearson correlation coefficient is $r = -0.85332$, highlighting the clear need for caution during feature selection on incomplete data. It is interesting to note that this behaviour is more subdued for the worst and median classification performers. This may indicate that with further stress on a higher imputation fitness metric, the dependence on completeness decreases. We, therefore, recommend the prioritized use of median range performers on the Pareto front rather than relying solely on strong classification.

5.1.2 *Performance of imputation module*

The final set of tests aims to interpret the performance of the imputation module, particularly in its ability to preserve the distributional characteristics across features. Whilst it is true that the imputation by design looks at a multidimensional feature space, it is important that the resultant univariate distributions for all features are inherently preserved. This may depend on several factors, such as the feature's individual completion rate, the collective data set's completion rate (and, by proxy, the contribution of other features' emptiness), as well as the available sample size. To this end we attempt to constrain the imputation model's performance based on the feature completion rate, such that a lower threshold of completion for features may inform future implementation of this methodology.

The principle idea behind the test is that, for all numerical features, their univariate distribution within the imputed data set is drawn from the same population as the sample data set. To ensure that no biases are introduced by any inherent statistically significant discrepancies between the host and comparison samples, the test handles both subsets, separately. The sample data set, made up of all instances with an observational value for that feature and henceforth referred to as the *initial feature-complete data set*, is first drawn from the full data set. Then, a random subsample is drawn, the size of which is defined by a variable feature-specific completion rate. The values for the feature being tested are subsequently removed from the subsample. The imputation model is then used to impute the resultant *feature-incomplete data set* to obtain an *imputed* feature-complete data set. Finally, a Kruskal–Wallis (K–W) non-parametric test is used the compare the initial and imputed data sets, where the null hypothesis states that the difference between the medians of both samples is insignificant. The non-normal tails in the distributions across all features led to the selection of the K–W over the normality-assuming ANOVA test. We test the null hypothesis at 2σ and 3σ, upon which it is rejected for the alternate hypothesis that the difference is significant. Hence, for progressively lower completion rates, this test is done to ensure that the null hypothesis remains accepted, up to the point where the K–W statistic's p-values are low enough for it to be rejected. The completion rate at this point would then serve as a threshold completion rate for the feature, which when viewed within the context of the data set's general completion rate and the available sample size, can inform feature and sample selection in future work.

To ensure that the K–W statistic is stable and not strongly biased from the random sampling used to 'empty' the data set, we perform the sampling, subsequent imputation and K–W test for a number of iterations. An initial expectation is that the number of iterations should depend on the available sample size for the test. We, therefore, subdivide all numeric features (i.e. all except the *disc* Galactic feature) into five sample size bins, for both the host sample and comparison sample.

To select the optimal number of iterations for each sample-size bin, we perform a test based on the variation of K–W statistic. For every bin, the full test is done for every feature for 20 iterations with a constant feature-specific completion rate of 90 per cent, and all features in a logarithmic scale are converted to linear scales so that the scale in the statistic can be stable across all features. Focusing on one feature at a time, the standard deviation of the K–W statistic can then be calculated for a progressively increasing number of iterations. A final trend was obtained by calculating the average K–W statistic deviation across all features within that particular bin, such that it could then be plotted as shown in Figs 12 and 13.

The number of iterations necessary for each sample bin was selected based on the visual interpretation of the trends in the plots. As the point of this preliminary test is to validate the choice of iterations, we merely select the lowest number of iterations beyond which the deviation in the statistic begins to plateau. The resultant number of iterations for all sample-size bins is presented in Table 11.

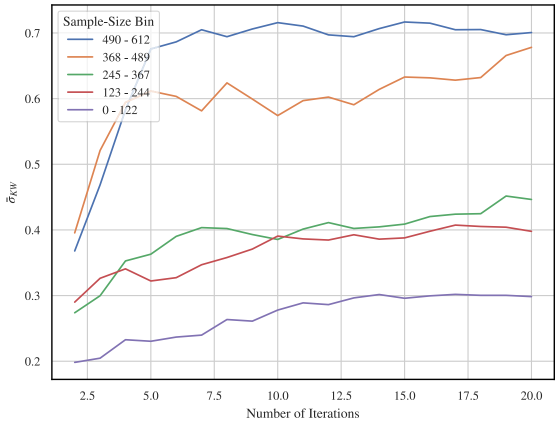

Figure 12. The average K–W statistic deviation against the number of iterations for the host sample. The point at which the trends began to plateau is taken as an indication of the minimum acceptable number of iterations for that sample-size bin.

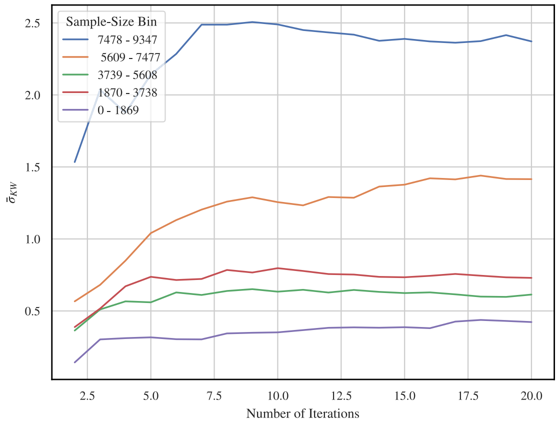

Figure 13. The average K–W statistic deviation against the number of iterations for the comparison sample. The point at which the trends began to plateau is taken as an indication of the minimum acceptable number of iterations for that sample-size bin.

Table 11. The selected number of iterations for all sample-size bins for the imputation preservation test.

Hosts sample size	Iterations	Comparison sample size	Iterations
0 – 122	15	0 – 1869	12
123 – 244	15	1870 – 3738	12
245 – 367	15	3739 – 5608	10
368 – 489	15	5609 – 7477	20
490 – 612	10	7478 – 9347	10

We present the minimum threshold values per feature in each data set in the `thresholds_Binary_implementation` directory provided in the accompanying GitHub repository.[10] The reason behind presenting the values in a table format, rather than represent them graphically, is due to the fact that we do not find any coherent, stable, nor statistically significant trend with data set or feature completeness across the features and data set variants. With the current configuration for the MOO methodology and chromosome selection, optimization does not lead to a stable sample of Pareto

[10] https://github.com/miguel-zammit-uom/MOO-of-Incomplete-Stellar-Data-for-Exoplanet-Hosts/

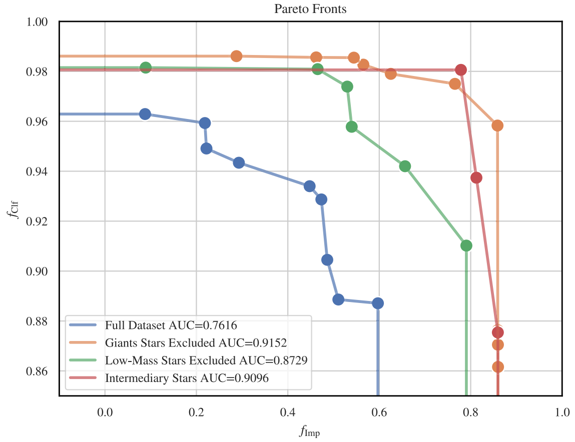

Figure 14. The Pareto front for all sample variants, with the corresponding AUC score provided in the legend.

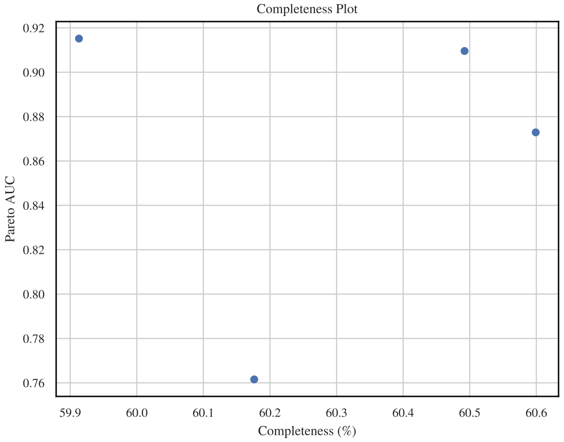

Figure 15. A plot of the AUC score against the completion rate for all sample variants. With a Pearson correlation coefficient of $r = 0.0647$, there does not seem to be evidence for a correlation between the completion rate and AUC. This may be mainly due to the fact that all four variants tend to have similar completion rates.

solutions which are necessarily adept at handling univariate distributional preservation at lower completion rates. This highlights the need for future work and applications to incorporate tests such as this within the framework of the MOO algorithm, perhaps as their own independent fitness functions or integrated within a single composite imputation fitness function. There are however certain characteristics one can highlight within the performance seen. There tends to be a substantially more sensitive dependency on the completeness of the features which were not logarithmic (which in the case of this data set it is all physical and galactic features except log g and disc location) than there is for logarithmic features. Our results show that for distributional preservation within the context of the imputation of data sets of the same nature as the Hypatia Catalog, linear features require higher completion rates to not deviate entirely from their natural distribution once the signal is injected through imputation. This is a natural byproduct of expressing features on a logarithmic scale, yet none the less, it remains an important factor to keep in mind during feature and sample selection. A second trend is the overall greater allowance for incompleteness in the comparison samples than in the hosts. This is expected to be mainly due to their initial sample sizes, so the label may likely not be consequential in this regard.

5.2 Sample variation

5.2.1 Pareto fronts and classification

As was the case for the feature variants, a closer look at the Pareto fronts and the classification scores of the three selected chromosomes per variant may present some insight into the dependency of performance on the type of stars present within the sample.

The Pareto fronts for the sample variation may provide a more interesting set of results with regard to the trends within the fitness space. It should be noted at this stage that the completion rates of all four variants are similar to the point where any discrepancies can be considered insignificant. Therefore, variation in AUC and shape of the fronts can more definitively provide context on which samples lend themselves to better performance. As can be seen from the fitness-space plot in Fig. 14, the best two performers are the *intermediary stars* and *giant stars excluded* data sets, followed by the *low-mass excluded* data set, and, by some distance, the *full data set*. This suggests that selection criteria based on star-type improves performance of both the imputation and classification modules, and further corroborates previous finding from the literature that the correlations between stellar host properties and the presence of planetary companions diminish for giant stars (Maldonado et al. 2013). The inhibited performance of the median classifier of the *intermediary stars* variant can be explained by seeing the relatively low sample size of its Pareto front. All four chromosomes tend to be in the high-imputation fitness region of the fitness space, populating a region in which a strong drop-off in classification scores is expected. Hence, further exploration of the fitness space may be needed to ascertain whether it outperforms the *giant stars excluded* data set. The *low-mass stars excluded* variant shows that whilst their presence in the data set does not inhibit classification performance, it does negatively affect the ASW of the fuzzy clustering of the imputation module. This can be due to the fact that the ‘host-defining’ signal for giant planets will be less polluted for the intermediary stars data sets due to the presence of a small number of giant planetary companions around low-mass stars despite the inhibited formation rates expected for giant planets around low-mass stars. The sample size is too small to reliably to ascertain whether or not a correlation is present, but a Pearson correlation coefficient of $r = 0.0647$ and insignificant change in completion rates suggests that it is very unlikely that the change in AUC is linked with completeness, as visually demonstrated in Fig. 15.

Reverting focus back to classification metrics in Table 12, an interesting feature of the best classifiers is the stronger performers of the sample variants from the full data set, with a significant leap in recall and F_1. This may be further evidence of the classification module’s greater optimization of its ability to detect host stars in the sample when noise in the form of low-mass stars, and to a greater extent giant stars, is removed from the data set. Similar to the investigation of the AUC score’s correlation with completion rates, in Fig. 16, we plot the maximum accuracy against the completion rate for all sample variants. Again, the presence of a correlation cannot be significantly determined due to the small sample size. The Pearson correlation coefficient, however, equals $r = 0.0290$, so it is very unlikely that the improvement in the maximum accuracy is linked with completeness.

6. MULTILABEL CLASSIFICATION

As mentioned in our methodology, the binary classification optimization and performance were used as a platform to implement and

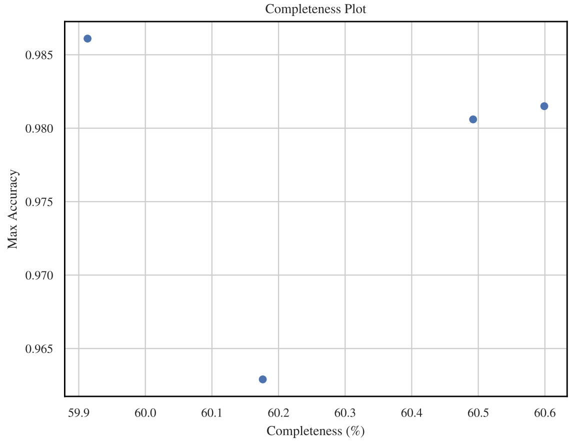

Figure 16. A plot of the maximum accuracy against the completion rate for all sample variants. With a Pearson correlation coefficient of $r = 0.0290$, there does not seem to be evidence for a correlation between the completion rate and the maximum accuracy. This may be mainly due to the fact that all four variants tend to have similar completion rates.

assess the performance on a multilabel classifier implementation. The apparent signal injection which was present within the feature variants for the binary classifier suggests that it should be present to a stronger degree in this case. Yet, the choice of four fitness functions, rather than the 2D case used for binary classification, may lead to a more subtle imputation module capable of capturing a more complete representation of the data set without necessarily overpowering low-completion samples with the injected signal.

This section will follow the same format as that of the binary classification, starting with an interpretation of the change in performance across the feature variants then moving on to the sample variants, in terms of the Pareto fronts, classification, and imputation performance.

6.1 Feature variation

6.1.1 Pareto fronts and classification

Due to the classification of all three labels being now treated as three separate fitness functions within the scope of the optimization, the Pareto fronts will occupy a 4D fitness space over which the trade-off becomes more nuanced. Hence, the immediate expectation is that the number of chromosomes occupying the Pareto fronts will be substantially higher. Furthermore, interpreting the fronts and the trade-off between fitness functions becomes more challenging. Implementing an AUC metric in four dimensions (or volume under plane in this case) may not capture the nuances seen with respect to each target label. Hence, this section will aim to show the shape of the Pareto front projected onto the three classification-imputation planes, i.e. with the classification fitnesses each plotted against the imputation metric. A front-line is drawn on each expressing the non-dominated set within that particular projection, to help illustrate the trade-off with respect to the corresponding classification label and the imputation module. The plotted chromosomes on the front are in each case a subset of the final set of Pareto solutions, as they would be the chromosomes which are non-dominated within that particular projection. Any noted trends or characteristics which will be highlighted within this section will be made with the pre-emptive understanding that these will not on their own show a complete picture, but any inherent dependencies on feature selection which are shown here may none the less prove useful.

Focusing first on the giant planet host label projection shown in Fig. 17, it is worthwhile to evaluate the model's performance within the context of the binary label implementation, which can be found in Fig. 9. It can be clearly seen that the set of Pareto solutions found has significantly changed, with AUC scores of all variants except the *full data set at 25 per cent threshold* scoring lower than in the previous set of results. Whilst this may be construed as the multilabel implementation underperforming, the reader is reminded of the signal injection which is expected to reside within the data set after imputation. High classification performance may not inherently indicate 'better' performance of the optimization model overall, but is merely one facet. The goal is to maximize classification performance, but only for it to lead to final imputed data set which is not significantly warped through imputation. Looking at individual performances, the *Vol.+Lith.+Fe+Phys.* and *No Phys.* data sets have the most ideal trade-off on the project front for the giant label. It is interesting to note that once siderophiles were included to the *Vol.+Lith.+Fe+Phys.* variant, the model sees the worst trade-off between imputation and giant planet classification. It is, however, when all variants are collectively seen within the context of the data set completion rate, where a stronger distinction can be made from the binary run. As can be seen in Fig. 18, the small dependency on data set completion for the Pareto front is no longer present in the projected front for the giant label with the Pearson correlation coefficient being half the magnitude. Now, it is important to restate that this is a projection onto an axis, so one cannot wholly conclude that the dependence is eliminated. However, it does shed light that whilst the inclusion of multiple labels may inject more information during imputation, the apparent influence of the giant planet host label may become regularized with the introduction, and equal prioritization, of other labels.

Moving onto the projected fronts of the low-mass planet host and multiplanet labels, similar trends can be seen as those found in the giant planet host label projection, with some marginal changes in rankings based on their AUC scores. As seen in Figs 19 and 20. The *Vol.+Lith.+Fe+Phys.* is now very marginally outperformed by the *No Phys.* data sets, but again the inclusion of siderophiles to the former sees a substantially less-optimal trade-off. For low-mass planets this is somewhat surprising, considering the literature's suggestions that their presence may indicate the accretionary budget for terrestrial planetary cores (Hinkel et al. 2019). Another change of note comes from the performance of the *All Chem.+Phys.* data set. For the low-mass planet host label, this variant, when seen relative to the other variant's front projections for all sets of labels, tends to achieve a better trade off than in the giant planet host and multiplanet label projections. This may be a slight indication that the inclusion of chemical species other than those specified in the other chemical group variants may boost performance with respect to the low-mass planet label.

Finally, and especially since the ranking remains relatively similar across all projections, the AUC scores' dependence on data set completeness follows suit from the giant planet host projection in that they show little-to-no dependence on data set completion, as shown in Figs 21 and 22. These, combined with that in Fig. 18, may be construed to make the case for the magnitude of the signal injection within the multilabel being more subdued. The reader should keep in mind however that these are only representations of a subset of the Pareto front, and that the true front resides within a 4D fitness space. Hence, any conclusions drawn need to be conservative. It may be true that within the context of the 2D projections there may be no

Table 12. Performance metrics of the set of highest, lowest, and median classification performers for each data set sample variant in their respective final Pareto front for the binary label implementation.

		Best classification performers				
Data set	Imputation fitness	Accuracy	Precision	Recall	F_1 score	ROC-AUC
Full data set	0.0877	0.9629 ± 0.0115	0.947 ± 0.0224	0.8455 ± 0.0494	0.8929 ± 0.0347	0.9174 ± 0.026
Giant stars omitted	0.2877	0.9861 ± 0.0034	0.9737 ± 0.0164	0.9505 ± 0.0138	0.9618 ± 0.0094	0.9723 ± 0.0067
Low-mass stars omitted	0.0891	0.9815 ± 0.0046	0.957 ± 0.0163	0.9425 ± 0.0253	0.9494 ± 0.0129	0.9664 ± 0.0121
Intermediary stars	0.7787	0.9806 ± 0.0049	0.9723 ± 0.0179	0.9212 ± 0.0187	0.9459 ± 0.014	0.9576 ± 0.0098
		Worst classification performers				
Data set	Imputation fitness	Accuracy	Precision	Recall	F_1 score	ROC-AUC
Full data set	0.5971	0.8871 ± 0.0123	0.8063 ± 0.0372	0.5085 ± 0.0647	0.6217 ± 0.0538	0.7405 ± 0.0321
Giant stars omitted	0.8598	0.8616 ± 0.0079	0.7581 ± 0.0505	0.3679 ± 0.0474	0.4929 ± 0.0441	0.6704 ± 0.0219
Low-mass stars omitted	0.7906	0.9102 ± 0.0108	0.7843 ± 0.0459	0.7109 ± 0.0379	0.7446 ± 0.029	0.8331 ± 0.0187
Intermediary stars	0.8594	0.8754 ± 0.0164	0.772 ± 0.0683	0.4603 ± 0.0642	0.5748 ± 0.0615	0.7147 ± 0.0337
		Median classification performers				
Data set	Imputation fitness	Accuracy	Precision	Recall	F_1 score	ROC-AUC
Full data set	0.4477	0.934 ± 0.0083	0.8952 ± 0.0367	0.7279 ± 0.0329	0.8022 ± 0.0251	0.8542 ± 0.0165
Giant stars omitted	0.7656	0.975 ± 0.0054	0.9532 ± 0.0167	0.9092 ± 0.0237	0.9305 ± 0.0154	0.9495 ± 0.012
Low-mass stars omitted	0.5300	0.9739 ± 0.0079	0.9561 ± 0.0202	0.8998 ± 0.0366	0.9267 ± 0.0231	0.9452 ± 0.0186
Intermediary stars	0.8125	0.9374 ± 0.0098	0.9163 ± 0.0321	0.7274 ± 0.0498	0.8098 ± 0.0331	0.8561 ± 0.0244

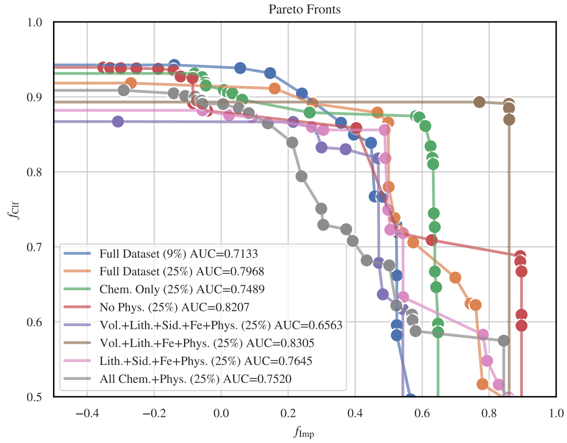

Figure 17. The projected Pareto front for the giant planet host label, for all feature variants in the multilabel implementation. The corresponding AUC score are provided in the legend.

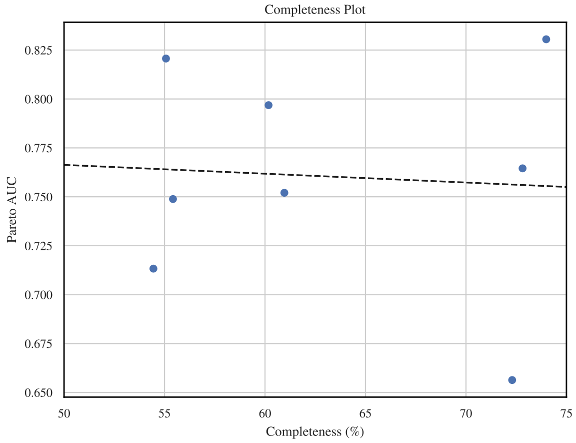

Figure 18. A plot of the AUC score against the completion rate for all feature variants on the projected Pareto front for the giant planet host label. With a Pearson correlation coefficient of $r = -0.0668$, half of that seen in the binary label implementation, shows little to no dependency on data set completion.

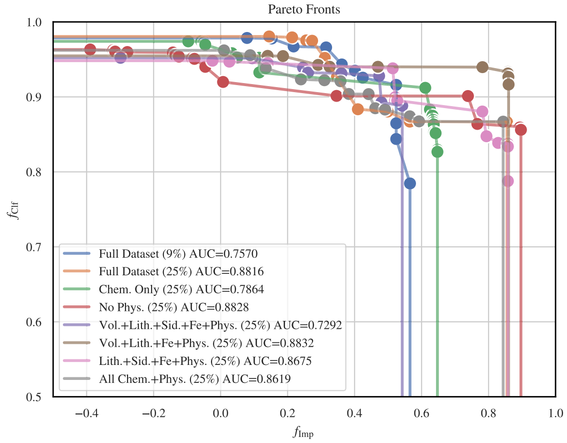

Figure 19. The projected Pareto front for the low-mass planet host label, for all feature variants in the multilabel implementation. The corresponding AUC score are provided in the legend.

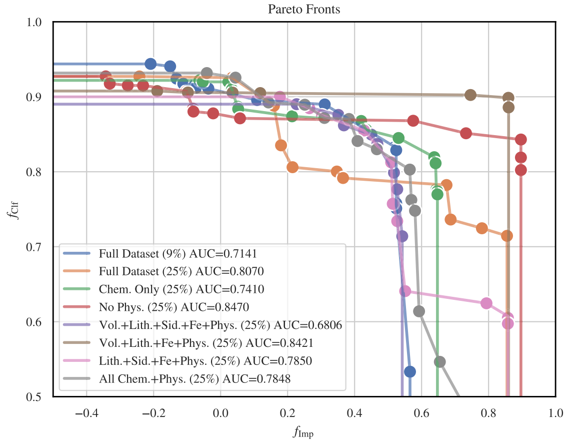

Figure 20. The projected Pareto front for the multiple planet host label, for all feature variants in the multilabel implementation. The corresponding AUC score are provided in the legend.

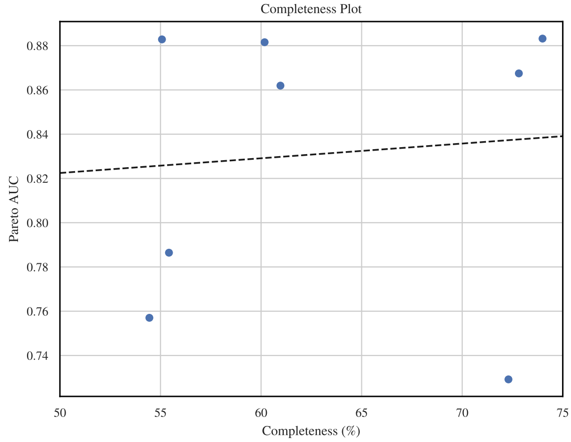

Figure 21. A plot of the AUC score against the completion rate for all feature variants on the projected Pareto front for the low-mass planet host label. With a Pearson correlation coefficient of $r = 0.0895$, shows little to no dependency on data set completion.

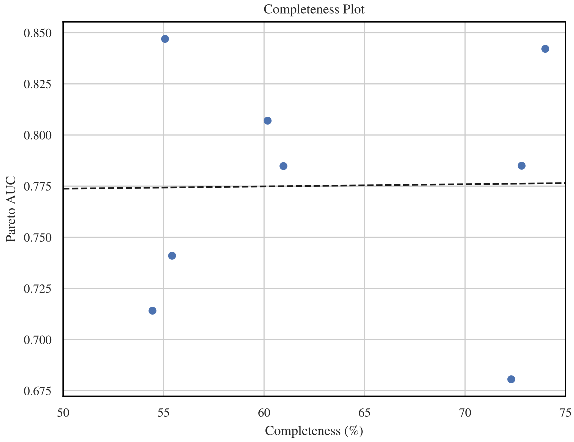

Figure 22. A plot of the AUC score against the completion rate for all feature variants on the projected Pareto front for the multiplanet host label. With a Pearson correlation coefficient of $r = 0.01559$, shows no dependency on data set completion.

inherent dependence on completion rates and that there is no apparent influence of the signal injected through greater degrees of imputation. However, it would not be prudent to extend this to a general statement regarding the model's performance in its optimization.

As was the case with binary implementation, classification scores for the multilabel classifier were investigated at individual points across the Pareto fronts for all feature variants. The methodology to select candidates for these tests may be somewhat less obvious in this case than it was previously for the binary label. We select our three candidates from each variant's Pareto front as follows: the mean of all three classification metrics, which in this case were the F_1 scores, was calculated. The chromosome with the highest mean F_1 is selected as the '*best*' performer, and the chromosome with the worst mean F_1 is the '*worst*' performer. The median performer is selected as the third candidate.

The reader may recall that in Table 10 for the single giant planet host label we present the accuracy, recall, precision, F_1, and ROC-AUC scores. To avoid an unnecessary load of information which may end up being more confusing than illuminating, we focus on providing the F_1 metric, which was used as the fitness function for this optimization, and the accuracy. The ROC curves for multilabel classification may lead to misinterpretation of the change in the TP rate with respect to the rate of false positive, especially in the case of inter-label dependencies which are obvious in this application. Furthermore, as the label distribution is imbalanced for all data set variations, cross-examining ROC-AUC scores may be misleading. This is a point which applies to the accuracy metric as well, which is why it was not implemented as a fitness metric in the first place, and should be kept in mind when evaluating performance through this metric. With regards to the omission of the precision and recall, since the F_1 score provides the harmonic mean of the two, and as it tends to be a stricter metric generally speaking, looking solely at the F_1 and accuracy for each label across the feature variants should be expected to provide a clear-enough picture for validation of performance of the classification module within its application in the algorithm.

Table 13 presents the classification scores of the multilabel implementation for all feature variants.

Focusing first on the top performers for each variant, it benefits the discussion to first consider the giant planet host label result, and to observe how they compare with the performance of the other two labels. All variants attain substantially high F_1 and accuracy metrics, with the *No Phys.*, *Chem. only*, and *full data set with 9 per cent threshold* data sets achieving the highest scores, albeit not reaching the heights of the binary implementation. This in of itself may be a good outcome, as the signal injection suspected in the previous result may be slightly more subdued with the introduction of multiple labels. The *Vol.+Lith.+Sid.+Fe+Phys.* data set shows the worst performance for the giant planet host label out of all the variants, which means that the variant rankings are similar to those obtained by the binary classifier implementation. This may in of itself be considered as a sanity check that the multilabel classifier is an extension of the binary classification, with the imputation module providing less reinforcement in the signal distinguishing giant planet hosts to comparison stars. Observing how performance extends to the other two labels, it can be seen that the rankings of which are the best performing variants have remained similar. For the multiplanet label, the performance only tends to show any change in the F_1 score, with all accuracy values remaining quite similar. Still, in both rocky and multilabel classification it is clear that stable generalization is occurring for all feature variants, with clear signal-injection maximizing the discernability between the label subsets.

The overall performance of the best classification chromosomes tends to suggest that at the high average classification – low imputation region of the fitness space, there tends to little variation on which labels respond best to specific feature groups. If a feature group achieves higher scores for one label, it will tend to do the same for the other two.

It is when the conversation switches to the worst performers that certain nuances come into play. Again, the giant planet host label performance is similar to that found for the binary classifier. The *No Phys.*, *Chem. only*, *Vol.+Lith.+Fe+Phys.*, and *All Chem.+Phys.* obtain the highest scores, with the *Vol.+Lith.+Sid.+Fe+Phys.*, *Lith.+Sid.+Fe+Phys.*, and full data set variants scoring significantly worse. However, it may be the case that the points of major interest for this set of runs come from the performance for the low-mass planet host label. It is apparent that even when the model underperforms for the giant planet host labelling, if it is provided with certain specific chemical groups it excels in low-mass planet host determination. In this case, it is likely that the contribution of the volatiles and lithophile groups contributed to greater response in the *Vol.+Lith.+Sid.+Fe+Phys.* and *Vol.+Lith.+Fe+Phys.* variants. While their contribution may be backed with theoretical work

Table 13. Performance metrics of the set of highest, lowest, and median classification performers for each data set feature variant in their respective final Pareto front for the multilabel implementation.

Best classification performers							
Data set	Imputation fitness	Giant planet host		Low-mass planet host		Multiple planet host	
		F_1 score	Accuracy	F_1 score	Accuracy	F_1 score	Accuracy
Full data set (25 per cent)	−0.2548	87.23 ± 1.97	95.50 ± 0.67	95.68 ± 1.15	98.05 ± 0.50	91.82 ± 1.72	97.91 ± 0.44
Full data set (9 per cent)	−0.1804	91.67 ± 1.67	97.03 ± 0.57	96.23 ± 1.25	98.29 ± 0.55	92.18 ± 1.40	97.99 ± 0.34
Chem. only	−0.0794	93.13 ± 1.19	97.52 ± 0.42	97.28 ± 0.76	98.76 ± 0.34	90.41 ± 1.64	97.58 ± 0.41
No Phys. (25 per cent)	−0.1425	93.56 ± 1.59	97.65 ± 0.58	95.93 ± 0.75	98.16 ± 0.32	89.48 ± 1.95	97.38 ± 0.49
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	−0.1302	84.04 ± 2.35	94.37 ± 0.85	92.82 ± 1.14	96.80 ± 0.49	87.43 ± 1.87	96.90 ± 0.42
Vol.+Lith.+Fe+Phys. (25 per cent)	−0.0604	88.18 ± 1.48	95.79 ± 0.54	94.14 ± 1.00	97.37 ± 0.43	90.35 ± 0.93	97.61 ± 0.20
Lith.+Sid.+Fe+Phys. (25 per cent)	−0.0617	87.63 ± 2.62	95.65 ± 0.88	93.56 ± 1.47	97.13 ± 0.63	86.41 ± 2.47	96.69 ± 0.57
All Chem.+Phys. (25 per cent)	−0.2932	89.32 ± 1.43	96.23 ± 0.51	94.36 ± 1.03	97.46 ± 0.44	87.73 ± 1.49	96.97 ± 0.38
Worst classification performers							
Data set	Imputation fitness	Giant planet host		Low-mass planet host		Multiple planet host	
		F_1 score	Accuracy	F_1 score	Accuracy	F_1 score	Accuracy
Full data set (25 per cent)	0.1804	46.15 ± 3.59	85.46 ± 0.93	86.88 ± 1.46	94.13 ± 0.66	52.56 ± 4.25	90.04 ± 0.73
Full data set (9 per cent)	0.5655	49.22 ± 3.68	82.89 ± 1.28	78.24 ± 2.08	89.99 ± 0.94	51.33 ± 2.92	87.98 ± 0.89
Chem. only	0.6375	72.15 ± 2.84	90.81 ± 0.90	84.28 ± 2.05	92.88 ± 0.89	33.81 ± 3.53	87.05 ± 0.74
No Phys. (25 per cent)	0.6285	70.89 ± 2.54	90.25 ± 0.89	86.37 ± 2.12	93.91 ± 0.88	45.20 ± 3.34	88.17 ± 0.54
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	0.4801	42.97 ± 3.76	85.16 ± 0.69	89.26 ± 1.75	95.27 ± 0.74	68.02 ± 3.13	92.76 ± 0.71
Vol.+Lith.+Fe+Phys. (25 per cent)	0.3255	54.44 ± 2.69	87.04 ± 0.75	94.02 ± 1.29	97.35 ± 0.55	65.88 ± 2.89	92.22 ± 0.49
Lith.+Sid.+Fe+Phys. (25 per cent)	0.8577	47.50 ± 4.61	81.46 ± 1.56	79.00 ± 1.78	90.25 ± 0.83	60.25 ± 2.66	89.55 ± 0.77
All Chem.+Phys. (25 per cent)	0.8436	55.73 ± 2.53	86.85 ± 0.79	86.67 ± 1.81	93.97 ± 0.76	39.32 ± 4.35	86.77 ± 1.01
Median classification performers							
Data set	Imputation fitness	Giant planet host		Low-mass planet host		Multiple planet host	
		F_1 score	Accuracy	F_1 score	Accuracy	F_1 score	Accuracy
Full data set (25 per cent)	0.4597	62.87 ± 3.60	89.14 ± 0.86	87.24 ± 1.69	94.32 ± 0.69	82.79 ± 2.36	95.99 ± 0.47
Full data set (9 per cent)	0.0568	93.27 ± 1.14	97.55 ± 0.40	90.09 ± 1.55	95.61 ± 0.62	74.20 ± 3.36	93.87 ± 0.72
Chem. only	0.4348	68.44 ± 3.45	89.55 ± 1.00	89.92 ± 1.17	95.41 ± 0.54	84.51 ± 1.86	96.11 ± 0.49
No Phys. (25 per cent)	−0.0664	78.89 ± 2.20	92.88 ± 0.61	92.22 ± 1.16	96.58 ± 0.47	82.09 ± 2.84	95.84 ± 0.62
Vol.+Lith.+Sid.+Fe+Phys. (25 per cent)	0.4690	81.20 ± 3.07	93.55 ± 0.96	86.99 ± 1.40	94.16 ± 0.62	67.37 ± 3.14	92.62 ± 0.70
Vol.+Lith.+Fe+Phys. (25 per cent)	0.1366	85.38 ± 1.98	94.95 ± 0.64	92.75 ± 1.25	96.80 ± 0.52	90.24 ± 1.57	97.59 ± 0.39
Lith.+Sid.+Fe+Phys. (25 per cent)	0.3934	76.77 ± 1.79	91.85 ± 0.67	88.11 ± 1.26	94.60 ± 0.56	75.45 ± 2.29	93.80 ± 0.54
All Chem.+Phys. (25 per cent)	0.3100	69.61 ± 2.96	90.10 ± 0.83	91.68 ± 1.61	96.21 ± 0.73	86.82 ± 2.17	96.72 ± 0.53

Table 14. The selected number of iterations for all sample-size bins for the imputation preservation test in the multilabel implementation.

Giant planet host label			
Hosts sample size	Iterations	Comparison sample size	Iterations
0 − 122	15	0 − 1869	12
123 − 244	15	1870 − 3738	12
245 − 367	15	3739 − 5608	10
368 − 489	15	5609 − 7477	20
490 − 612	10	7478 − 9347	10
Low-mass planet host label			
Hosts sample size	Iterations	Comparison sample size	Iterations
0 − 155	15	0 − 1836	12
156 − 310	15	1837 − 3672	12
311 − 465	15	3673 − 5508	10
466 − 620	10	5509 − 7344	20
621 − 775	10	7345 − 9181	10
Multiple planet host label			
Hosts sample size	Iterations	Comparison sample size	Iterations
0 − 90	15	0 − 1901	12
91 − 181	15	1902 − 3802	12
182 − 271	15	3803 − 5704	10
272 − 362	15	5705 − 7605	20
363 − 453	15	7606 − 9507	10

from the literature, future work should aim to establish whether it was truly their presence which led to higher performance. In this regard, the importance of siderophiles for planetary cores in terrestrial planets suggests that the same should have occurred for the *Lith.+Sid.+Fe+Phys.*, yet the variant's performance spike is much more subdued.

The median classification performers deviate from the trends seen with the extremal groups, albeit the performance from the giant planet host label is again a subdued yet similar performance to the binary implementation. The *full data set with 9 per cent threshold* obtains the highest scores for the giant planet host label, yet drops down the pecking order when it comes to the low-mass planet host and multiplanet labels. The opposite is true of the *Vol.+Lith.+Fe+Phys.* variant, for which the latter two labels see an uptick in performance after it attaining a mid ranking for the giant planet host determination. This may imply, when considered with the performances of the other sets of chromosomes, that for giant planet hosts, a more generalized chemical view may be needed whilst more specific subgroups of chemical species become more important at lower masses. This is an aspect of performance which will need to be validated/refuted once future work establishes a more robust implementation of this optimization model.

The results from the binary classifier implementation tended to imply a dependence of the accuracy scores on the data set completion rate, with a relatively strong anti-correlation detected. This was interpreted as an indication that with lower completion rates, a more

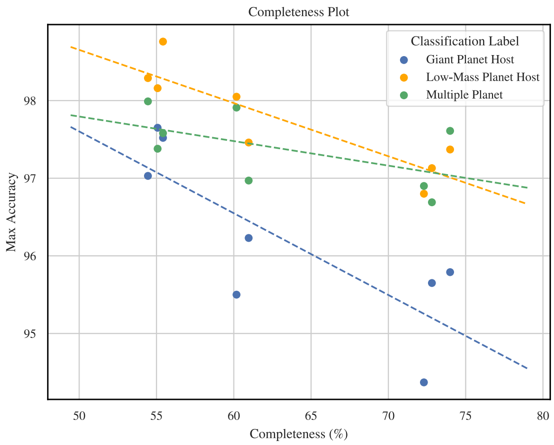

Figure 23. A plot of the maximum accuracy against the completion rate for all feature variants in the multilabel implementation.

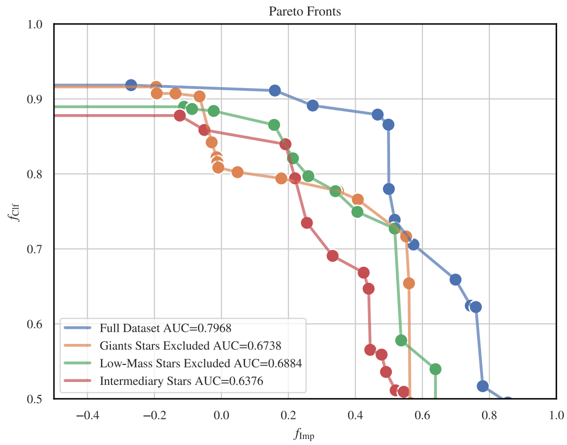

Figure 24. The projected Pareto front for the giant planet host label, for all sample variants in the multilabel implementation. The corresponding AUC scores are provided in the legend.

significant amount of signal-injected is expected to occur through the imputation module, which would imply that the maximal accuracy reached is increased. Extending this behaviour to the multilabel implementation, it would be valid to expect that with the inclusion of several labels, rather than just one, would limit the extent to which any individual label may have in injecting the data. We therefore carry out the same test on the maximum accuracies obtained for all three labels, as shown in Fig. 23. As can be seen from the trend lines, all three labels tend to show the presence of an inverse relationship, implying the same behaviour seen with the giant planet host label in the binary implementation. However, as seen from the slightly shallower slope than that for the binary classifier, the influence of the label can be seen to be slightly subdued. The Pearson correlation of $r = -0.7993$ is also slightly smaller. None the less, its presence still remains. The low-mass planet host label is has the highest correlation coefficient with $r = -0.8802$, whilst the multiple planet label scores a $r = -0.5606$ and the shallowest slope. It is clear that as was the case in the binary label implementation, signal injection is present, such that it should be kept in mind and regulated through managing the data set completion rate through feature selection. However, the nature of including multiple labels implies that the data set is provided with enough categorical diversity such that the impact of one single label is more subdued.

6.1.2 Performance of imputation module

Extending our work from the binary implementation, we carry out the same univariate distribution preservation test on the multilabel optimization model. The procedure for the test remains the same from the binary test, which is why this section will remain concise. The change in application comes from the fact that due to there being three binary labels rather than just one, the number of constrained thresholds triples. Furthermore, interpretation of the constraints needs to keep in mind that there will be an instilled co-dependency during imputation on all three labels, meaning that the performance of say, low-mass planet hosts versus their comparison sample, cannot be taken to be the constraint for the case if this were the sole label. The performance of imputation in this implementation will depend on the collective signal injection of all three. Hence these constraints may be used for feature selection and sample selection if these three labels are implemented.

Focusing back on the application of the preservation test, we select the number of iterations for each sample-size bin based on the initial exploration of the average standard deviation of the K–W statistic presented in Figs 12 and 13. Since for all hosts the sample ranges will be similar, we map the number of iterations to the sets of bins for the other two labels, as shown in Table 14. The resultant thresholds are provided in the `thresholds_Multilabel_implementation` directory in the accompanying GitHub repository.

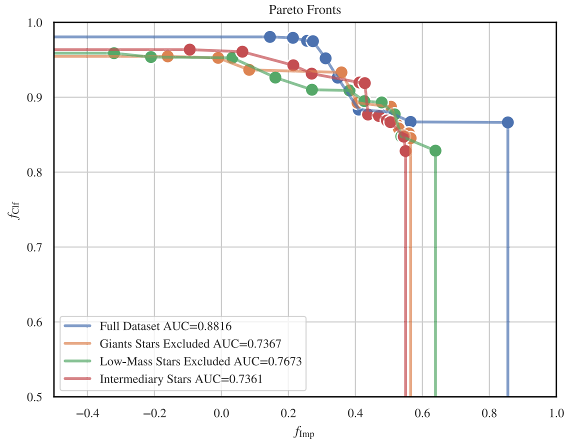

Figure 25. The projected Pareto front for the low-mass planet host label, for all sample variants in the multilabel implementation. The corresponding AUC scores are provided in the legend.

6.2 Sample variation

6.2.1 Pareto fronts and classification

Following the same procedure as that for the feature variants, and as was done for the binary implementation, the Pareto fronts for all four sample variant data sets were projected across the three classification-imputation planes.

As can be seen for the giant planet host label in Fig. 24, the relative performance changes again with respect to the binary run. Where before the full data set tended to be the clear worst performer, it obtains the highest AUC score by a significant margin. Inversely, the intermediary star sample became the worst performer. This trend extends to the other two projections in Figs. 25, 26 respectively, highlighting that when incorporating multiple labels

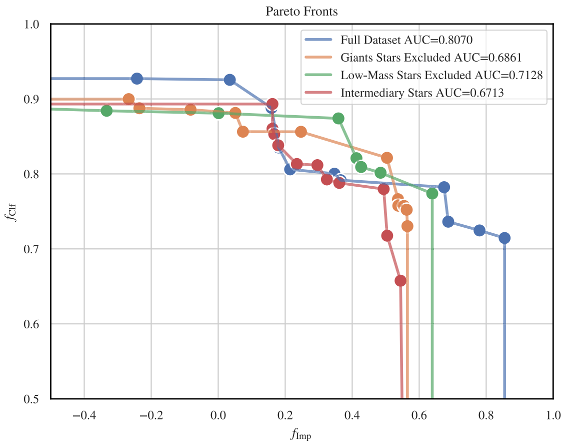

Figure 26. The projected Pareto front for the multiple planet host label, for all sample variants in the multilabel implementation. The corresponding AUC scores are provided in the legend.

for different planetary classes and configurations, a more diverse data set seems to intrinsically perform better with respect to the trade-off between imputation and classification. The performance differences between the low-mass and giant star exclusion data sets are almost indistinguishable. These trends may suggest that whilst sample selection may substantially aid performance when using binary label implementation, a diverse population of stars may suit multilabel implementations in providing a more complete picture.

We previously discussed whether or not a dependence on data set completion is present in the binary implementation. Whilst the Pearson correlation coefficient are stronger in this case, reaching $r = -0.2701$ for the giant planet hosts projection, the small sample size again makes any correlation inconclusive. As was the case before, the completion rates of all four are similar to the point where they are indistinguishable.

The classification metrics in Table 15 extend on those shown for the feature variation of the multilabel implementation by presenting the F_1 and accuracy metrics for all three labels.

Starting with the giant planet host label scores for the best classification performers, a similar trend emerges to that seen for the binary implementation. The data set without giant stars performs the best with a significantly higher accuracy and F_1. The intermediary stars data set does however drop-off to comparable, if not slightly worse scores than the full data set. It is interesting to note that when looking at low-mass planet host label performances, the data set without giant stars suffers a noticeable drop in performance from the full data set. This follows from literature that the giant stars do not exhibit the same correlations present in main sequence stars with the presence of giant planetary companions (Maldonado et al. 2013), such that their omission from the data set favours giant planet host detection more than it does for lower mass companions.

The two other sets of chromosomes tend to mirror the binary implementation's suggestions that the data set without giant stars, be it that being the sole exclusion or if the data set is composed of intermediary stars, performs better in distinguishing giant planet hosts from comparison stars. This can be seen occur to a small extent for the worst classification performers, and to a far more significant amount for the median performers.

Moving onto the low-mass planet host label, it seems likely that upon consideration of all three sets of chromosomes that performance is optimal either with the more diverse sample of stars as given by the full data set, particularly in the low imputation score region, or with the removal of low-mass stars from the sample, although this is inconclusive due to the inconsistency for the intermediary stars sample in the worst classification performers set of runs. The multiplanet label tends to benefit from having the full data set at its disposal, with only the intermediary stars sample for the median set of performers slightly outperforming its full data set counterpart.

7. DISCUSSION AND FUTURE WORK

In this body of work, we implement MOO and the power of ML to build an exoplanet host classifier whilst simultaneously imputing a large incomplete stellar abundance and parameter data set. The overarching design follows the implementation of Khorshidi et al. (2020), with some major changes to suit the nature of the data set and the classification task at hand. In the use-cases of the model, however, we diverge from the methodology proposed by Khorshidi et al., in that we present the performance of both the classification and imputation modules over several feature and sample variations of the data set as a method for attaining reliable imputation, rather than to train a classifier to be used with unseen data. Our tests aim to validate and constrain the use of the model in applications where it may be useful to inject a signal influenced by the target labels into the imputed data set in a nuanced way. This can then be used in comparison studies with complete, observational data such that samples may be seen within the context of the imputed data set to constrain label recommendation. As this is our first implementation, our goal was to constrain the methodology. We initially test the implementation of three different classification modules, and after confirming that XGBoost is the most stable performer out of the three, we apply a binary label implementation of the model, where we test its use in developing a model trained to detect giant planet hosts, and then extend this work to a multilabel classification model, in which labels for giant planet hosts, low-mass planet hosts, and multiplanet labels are used. For both cases, we constrain feature and sample selection, as well as the strengths and weakness of the implementation module.

The first section of results revolves around the selection of which of the three selected options for the classification module should be implemented for this particular use-case and data set. We select an initial feature set based on a 35 per cent threshold for the feature completion rate, which is the highest used in this work, to ensure that the focus for these sets of results is placed on the performance of the classification module and there is less bias introduced through the signal injection from the imputation module. As their labelling tends to be the most confident, and to be able to more unambiguously assess performance on individual labels, the control task was selected to be the binary classification of giant planet hosts. We find that from our three implementations and within the corresponding hyperparameter ranges which were tested, the XGBoost algorithm achieves the highest performance metrics, whilst also tending to be the most stable and precise, upon consideration of the magnitude of the uncertainties. The Pareto front achieved by the MOO algorithm with the XGBoost implementation also proves to be significantly more optimal than the other two, especially with respect to the AUC scores. This led to the decision that the XGBoost should be chosen to be implemented as the classifier within the classification module of the algorithm. It should be noted that the ANN implementation does tend to achieve relatively high classification scores, albeit with an apparent sharper trade-off with the imputation fitness. We recommend that future work should test different network architectures and regularization techniques such that its performance can become more comparable with the XGBoost.

Table 15. Performance metrics of the set of highest, lowest, and median classification performers for each data set sample variant in their respective final Pareto front for the multilabel implementation.

		Best classification performers					
Data set	Imputation fitness	Giant planet host		Low-mass planet host		Multiple planet host	
		F_1 score	Accuracy	F_1 score	Accuracy	F_1 score	Accuracy
Full data set	−0.2548	87.23 ± 1.97	95.50 ± 0.67	95.68 ± 1.15	98.05 ± 0.50	91.82 ± 1.72	97.91 ± 0.44
Giant stars omitted	−0.1950	91.61 ± 1.49	97.34 ± 0.49	93.65 ± 1.39	96.89 ± 0.65	86.73 ± 2.08	96.60 ± 0.50
Low-mass stars omitted	−0.1988	85.71 ± 1.83	94.98 ± 0.56	94.06 ± 1.22	97.38 ± 0.51	87.32 ± 1.82	96.87 ± 0.40
Intermediary stars	−0.1641	85.93 ± 1.94	95.74 ± 0.55	91.89 ± 1.33	96.09 ± 0.62	85.71 ± 2.12	96.47 ± 0.50
		Worst classification performers					
Data set	Imputation fitness	Giant planet host		Low-mass planet host		Multiple planet host	
		F_1 score	Accuracy	F_1 score	Accuracy	F_1 score	Accuracy
Full data set	0.1804	46.15 ± 3.59	85.46 ± 0.93	86.88 ± 1.46	94.13 ± 0.66	52.56 ± 4.25	90.04 ± 0.73
Giant stars omitted	0.5293	40.72 ± 5.30	86.92 ± 0.79	85.57 ± 1.19	92.77 ± 0.60	40.23 ± 5.41	87.38 ± 0.89
Low-mass stars omitted	0.4785	50.54 ± 3.52	83.85 ± 1.06	89.08 ± 1.15	95.09 ± 0.52	43.74 ± 2.75	86.73 ± 0.74
Intermediary stars	0.5498	43.55 ± 4.23	86.96 ± 0.67	82.42 ± 1.92	91.17 ± 0.93	45.92 ± 4.74	88.74 ± 0.65
		Median classification performers					
Data set	Imputation fitness	Giant planet host		Low-mass planet host		Multiple planet host	
		F_1 score	Accuracy	F_1 score	Accuracy	F_1 score	Accuracy
Full data set	0.4597	62.87 ± 3.60	89.14 ± 0.86	87.24 ± 1.69	94.32 ± 0.69	82.79 ± 2.36	95.99 ± 0.47
Giant stars omitted	0.4063	76.04 ± 2.65	93.11 ± 0.73	89.04 ± 1.66	94.63 ± 0.79	74.91 ± 2.34	93.73 ± 0.56
Low-mass stars omitted	0.1424	71.24 ± 2.70	90.59 ± 0.84	88.49 ± 1.21	95.00 ± 0.48	78.75 ± 2.50	95.12 ± 0.52
Intermediary stars	0.1621	69.42 ± 3.04	91.28 ± 0.82	91.24 ± 1.39	95.73 ± 0.63	85.71 ± 2.92	96.41 ± 0.71

The next two sets of results use the selected implementation to test the algorithm's performances in two use-cases, one where it is implemented in terms of binary classification of giant planet hosts, and in terms of multilabel classification of giant or low-mass planet hosts, and planet multiplicity. The aim of this comparison is to evaluate whether a set of Pareto solutions in either case may provide a sample of chromosomes which provide reliable imputation. In particular, the exploration of the trade-off of the extent of the signal injection and resultant high classification scores with the selection of features, samples and, by extension, the completion rates of the data set. We find that when testing feature variation in both implementations, signal injection tend to have an anti-correlation with the completion rate of the data set. The less complete the data set, the greater the influence of the target labels during imputation. The Pareto front does show a small but non-negligible trend through their respective AUC metric, yet it is in the high-classification/low-imputation region of the fitness space for each variant that the anti-correlation becomes significantly strong. Hence, it is important that future work keeps this factor in mind during feature selection, as the extent of the signal injection will depend on the particular use-case. A case can be made for our methodology being applied to uses where signal injection needs to be kept at a minimum to preserve the intrinsic nature of the data set and should therefore aim for a high completion rate. A case may also be made for uses where comparative analysis to secondary data sets may require a more significant and pronounced injected from the target labels, to maximize the expected signal, and would therefore benefit from a greater selection of features at the expense of completion rate. This all being said, an interesting result when comparing the multilabel performance with the classification performance is that the strength and slope of the anti-correlation for the giant planet host label becomes more subdued when other labels are included. This makes sense from the aspect of imputation, as other, sometimes conflicting signals are injected into the data set, therefore leading to a final imputation setting which is not solely optimized to lead to an optimal giant planet host classifier, but rather an all-rounder scoring well for all three labels. The binary classifier does tend to score higher than the multilabel classifier in terms of giant planet host determination, yet does not do so to a significantly larger amount which would outweigh this minor alleviation of overfitting during imputation. We therefore recommend the implementation of multilabel classification such that the resultant signal-injection becomes more diverse and representative of a more complete picture of exoplanet host demographics. It is important to reiterate a point mentioned in Section 2.2 that the inclusion of a planet multiplicity label may result in the introduction of strong observational biases with the current known population of multiplanet host stars. It was kept as a target feature in this instance mainly due to the aim of constraining its use in medium or long-term future work, at which point alternative detection techniques other than transit photometry and radial velocities may provide a more substantial contribution to the exoplanet population used in the data set. Hence, we recommend that whilst the implementation of multilabel classification should be used, we also suggest that the multiplicity label is not yet incorporated.

As a complementary note to the process of selecting features and/or samples, further constraints can be introduced in future iterations of the model for which the associated error with the abundances are included as additional selection criteria. The current imputation does not differentiate between well and poorly constrained abundance values, which therefore should be expected to inhibit model performance.

Comparing the overall results across the three labels in the multilabel implementation, it is the two labels pertaining to the specific exoplanet classes (i.e. the giant planet and low-mass planet host labels) that show significant trends in performance. Whilst the multiple planet label does tend to show some feedback response from imputation, classification, and the overall optimization of the MOO algorithm, its significantly more erratic performance metrics does not suggest consistent generalization over the metrics. We therefore refrain from postulating on any implications from the results.

Moving onto the performances of the feature variants, we find that feature selection may become particularly pertinent for different labels. It is important that this interpretation remains conservative and not be taken to be conclusive, but these initial results suggest that giant planet host classification tends to perform better with a more general selection of chemical species, whilst low-mass planet host determination performs better when focusing on selective chemical groups. It is unclear why certain groups, namely the siderophiles, which are expected and shown to be important for core formation in terrestrial planets do not lead to a positive response from the model. Future work should investigate the reason why this occurs and establish ideal combinations for all labels.

Sample variation highlighted certain interesting trends which again were dependent on the label being considered. Giant planet host classification metrics across the different sample sets showed an improved performance once giant stars were excluded from the data set, which is a result corroborated by the literature (Maldonado et al. 2013). This trend disappears for lower mass planets, further implying that this omission mainly benefits giant planet host determination. For the lower mass planet host and multiplicity labels, it seems that more diverse data sets tend to perform well, although excluding low-mass stars from the sample may benefit the performance of the former label. We aim to investigate and constrain these characteristics further in future work

The final set of tests that should be re-mentioned is the constraints set on the imputation module vis-a-vis the feature completion rate with respect to the complete data set completion and sample size. This was tested based on the idea that the imputation should ideally preserve the univariate distributions of each feature in the data set, for which we find the minimum feature completion rate for the distributions of the imputed and complete data set accepts the null hypothesis of the K–W test, at 2σ and 3σ. We present this as a resource provided in the linked GitHub repository as tables, for both the binary and multilabel implementation. Future work will aim to engage in more exhaustive tests to more fully constrain the use of the imputation module within the MOO algorithm. That being said, any potential use-case would benefit from consulting these thresholds during feature selection.

Evaluating the performance of the imputation module across all the runs, it may be the case that the chosen ASW metric does not in of itself lead to a fully reliable imputation module. This was expected of course, as the classification metrics themselves are a measure of performance of the imputation as well as the classification. However, it seems beneficial that future work should incorporate more imputation metrics which test the preservation of distributional characteristics either with a univariate or multivariate approach. It would require further tests to determine whether such alternative fitness functions would work best instead of or in addition to the ASW metric.

All of this being said, the true test of this implementation will be in its practical use. As mentioned in the motivation, one of the use-cases which for this work may be in host star recommendation. Using the imputed data set's distributional characteristics to assess the likelihood of stars in a holdout comparison sample being incorrectly labelled may lead to a potential shortlist of stars which may have the chemical markers to suggest that it is likely that a planetary companion had formed in its system. Such a method would be independent of detection sensitivity limits or orbital configuration, yet would allow for the recommendation of a shortlist of stars for follow-up observations.

The immediate next steps following this work can be summarized as follows:

(i) Select alternative imputation fitness functions and test their use both instead of and in addition to the ASW metric.

(ii) Test whether or not the inclusion of the associated uncertainties of the features during the selection methodology positively affects performance.

(iii) Test different network architectures and regularization techniques in the ANN model implementation.

(iv) Further constrain the dependencies and discrepancies of low-mass and giant planet host classification on the chemical species selected as features.

(v) Further constrain the dependencies and discrepancies of low-mass and giant planet host classification on the selected star sample in the data set.

(vi) Further constrain the dependencies on feature and data set completion rates for the imputation module.

(vii) Apply the model in a comparative analysis for host star recommendation.

8. CONCLUSION

In conclusion, this body of work attempts to implement an MOO algorithm and the predictive power of ML to build a model which results in reliable imputation of incomplete data sets through the guiding hand of ML classification. Said reliable imputation could then be used in several use-cases where comparative analysis requires a complete representation of the data set, being especially useful in the case of notoriously incomplete chemical species. The performance shown here provides promising results for the first iteration of this work with several aspects of its implementation being adequately constrained. Improving on these constraints and the selection of fitness functions, feature and sample selection and classifier configuration will be the next step, after which the model may have the potential to be a useful tool in host star recommendation and the exploration of the star–planet connection.

DATA AVAILABILITY

The data underlying this article were accessed and compiled from the Hypatia Catalog data base in May 2023. The catalogue may be freely accessed through the appropriate API and website as described in Hinkel et al. (2014). The auxiliary thresholds result for use of our design may be found in https://github.com/miguel-zammit-uom/MOO-of-Incomplete-Stellar-Data-for-Exoplanet-Hosts/. The code implementation of our model may be shared upon reasonable request to the corresponding author.

ACKNOWLEDGEMENTS

The authors would like to thank the Institute of Space Sciences and Astronomy and the Faculty of Science at the University of Malta for providing access to computational facilities at the Ultrawideband Laboratory. We would also like to thank the referee for their helpful comments on the work presesnted here.

The research shown here acknowledges the use of the Hypatia Catalog data base, an online compilation of stellar abundance data as described in Hinkel et al. (2014), which was supported by NASA's Nexus for Exoplanet System Science (NExSS) research coordination network and the Vanderbilt Initiative in Data-Intensive Astrophysics (VIDA). This research has also made use of the National Aeronautics and Space Administration (NASA) Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the

Exoplanet Exploration Program. Finally, this research has made use of data obtained from or tools provided by the portal exoplanet.eu of The Extrasolar Planets Encyclopaedia.

REFERENCES

Adibekyan V. Z., Delgado Mena E., Sousa S. G., Santos N. C., Israelian G., Hernández J. I. G., Mayor M., Hakobyan A. A., 2012a, A&A, 547, A36
Adibekyan V. Z. et al., 2012b, A&A, 543, A89
Adibekyan V. et al., 2021, Science, 374, 330
Alves S., Do Nascimento J., De Medeiros J. R., 2010, MNRAS, 408, 1770
Bakos G. Á., Lázár J., Papp I., Sári P., Green E. M., 2002, PASP, 114, 974
Bezdek J. C., 2013, Pattern Recognition with Fuzzy Objective Function Algorithms. Springer, New York
Bodaghee A., Santos N. C., Israelian G., Mayor M., 2003, A&A, 404, 715
Boley K., Christiansen J., 2023, American Astronomical Society Meeting Abstracts, 55, 430.03
Bond J. C. et al., 2008, ApJ, 682, 1234
Borucki W. J., 2016, Rep. Prog. Phys., 79, 036901
Borucki W. J. et al., 2011, ApJ, 736, 19
Brady M. T., Bean J. L., 2022, AJ, 163, 255
Buchhave L. A., Latham D. W., 2015, ApJ, 808, 187
Burn R., Schlecker M., Mordasini C., Emsenhuber A., Alibert Y., Henning T., Klahr H., Benz W., 2021, A&A, 656, A72
Chen T., Guestrin C., 2016, Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min. (KDD'16), XGBoost: A Scalable Tree Boosting System. Assoc. Comput. Mach., New York, p. 785
Clanton C., Gaudi B. S., 2014, ApJ, 791, 91
Cortes C., Vapnik V., 1995, Mach. Learn., 20, 273
Dai Y.-Z., Liu H.-G., An D.-S., Zhou J.-L., 2021, AJ, 162, 46
Deb K., Pratap A., Agarwal S., Meyarivan T., 2002, IEEE Trans. Evolut. Comput., 6, 182
Delgado Mena E. et al., 2014, A&A, 562, A92
Delgado Mena E. et al., 2015, A&A, 576, A69
Emmerich M. T., Deutz A. H., 2018, Nat. Comput., 17, 585
Fischer D. A., Valenti J., 2005, ApJ, 622, 1102
Gilli G., Israelian G., Ecuvillon A., Santos N. C., Mayor M., 2006, A&A, 449, 723
Gonzalez G., 1997, MNRAS, 285, 403
Gonzalez G., 2008, MNRAS, 386, 928
Gonzalez G., 2009, MNRAS, 399, L103
Gonzalez G., 2014, MNRAS, 443, 393
Gonzalez G., 2015, MNRAS, 446, 1020
Gonzalez G., Laws C., 2000, AJ, 119, 390
Haywood M., 2008, A&A, 482, 673
Haywood M., 2009, ApJ, 698, L1
Hinkel N. R., Timmes F. X., Young P. A., Pagano M. D., Turnbull M. C., 2014, AJ, 148, 54
Hinkel N. R., Unterborn C., Kane S. R., Somers G., Galvez R., 2019, ApJ, 880, 49
Ida S., Lin D., 2005, Prog. Theor. Phys. Supp., 158, 68
Ioffe S., Szegedy C., 2015, Proc. 32nd International Conference on Machine Learning, Vol. 37. PMLR, Lille, France, p. 448
Israelian G., Santos N. C., Mayor M., Rebolo R., 2004, A&A, 414, 601
Johnson J. A., Apps K., 2009, ApJ, 699, 933
Johnson J. A., Aller K. M., Howard A. W., Crepp J. R., 2010, PASP, 122, 905
Khorshidi H. A., Kirley M., Aickelin U., 2020, Proc. 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, Glasgow, UK, p.1
Kotsiantis S., Kanellopoulos D., Pintelas P., 2006, GESTS Int. Trans. Comput. Sci. Eng., 30, 25
Lanza A., 2010, A&A, 512, A77
Lineweaver C. H., 2001, Icarus, 151, 307
Lissauer J. J., Dawson R. I., Tremaine S., 2014, Nature, 513, 336
Lodders K., Palme H., Gail H., 2009, JE Trümper, 4, 44
Mah J., Bitsch B., 2023, A&A, 673, A17
Maldonado J., Villaver E., Eiroa C., 2013, A&A, 554, A84
Mann A. W., Gaidos E., Kraus A., Hilton E. J., 2013, ApJ, 770, 43
Paulson D. B., Yelda S., 2006, PASP, 118, 706
Pepper J. et al., 2007, PASP, 119, 923
Perryman M., 2018, The Exoplanet Handbook. Cambridge Univ. Press, Cambridge
Petigura E. A. et al., 2018, AJ, 155, 89
Pollacco D. L. et al., 2006, PASP, 118, 1407
Reid I. N., 2002, PASP, 114, 306
Ricker G. R. et al., 2014, J. Astron. Telesc. Instrum. Syst., 1, 014003
Sandford E., Kipping D., Collins M., 2019, MNRAS, 489, 3162
Santos N. C., Israelian G., Mayor M., Bento J. P., Almeida P. C., Sousa S. G., Ecuvillon A., 2005, A&A, 437, 1127
Santos N. C. et al., 2017, A&A, 603, A30
Sousa S. G. et al., 2008, A&A, 487, 373
Sozzetti A., 2004, MNRAS, 354, 1194
Zammit M. A., Zarb Adami K., 2024, MNRAS, 527, 9930
Zammit M. A., Zarb Adami K., 2023, Machine Learning for Astrophysics: Proc. ML4Astro International Conference, Vol. 60. Springer International Publishing, Cham, p. 111
Zhu W., Dong S., 2021, ARA&A, 59, 291

This paper has been typeset from a TEX/LATEX file prepared by the author.