RAS Techniques and Instruments

RASTAI **2,** 695–709 (2023)
Advance Access publication 2023 November 2

https://doi.org/10.1093/rasti/rzad050

Lessons learned from the 1st Ariel Machine Learning Challenge: Correcting transiting exoplanet light curves for stellar spots

Nikolaos Nikolaou,[1★] Ingo P. Waldmann,[1] Angelos Tsiaras,[1,2] Mario Morvan,[1] Billy Edwards,[1] Kai Hou Yip,[1] Alexandra Thompson,[1] Giovanna Tinetti,[1] Subhajit Sarkar,[3] James M. Dawson,[3] Vadim Borisov,[4] Gjergji Kasneci,[4] Matej Petković,[5] Tomaž Stepišnik,[5] Tarek Al-Ubaidi,[6,7] Rachel Louise Bailey,[7] Michael Granitzer,[8] Sahib Julka,[8] Roman Kern,[9] Patrick Ofner,[9] Stefan Wagner,[10] Lukas Heppe,[11] Mirko Bunse,[11] Katharina Morik[11] and Luís F. Simões[12]

[1]*Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK*
[2]*INAF – Osservatorio Astrofisico di Arcetri, Largo E. Fermi 5, I-50125 Firenze, Italy*
[3]*School of Physics and Astronomy, Cardiff University, The Parade, Cardiff CF24 3AA, UK*
[4]*Department of Computer Science, University of Tuebingen, Tuebingen 72076, Germany*
[5]*Jožef Stefan Institute, Ljubljana 1000, Slovenia*
[6]*DCCS GmbH, 8041 Graz, Austria*
[7]*Space Research Institute, Austrian Academy of Sciences, 8042 Graz, Austria*
[8]*Chair of Data Science, University of Passau, 94032 Passau, Germany*
[9]*Know-Center GmbH, Research Center for Data-Driven Business & Big Data Analytics, A-8010 Graz, Austria*
[10]*Commission for Astronomy, Austrian Academy of Sciences, 8042 Graz, Austria*
[11]*Lamarr Institute for Machine Learning and Artificial Intelligence, TU Dortmund University, Dortmund 44221, Germany*
[12]*ML Analytics, Lisbon, Portugal*

Accepted 2023 October 25. Received 2023 October 19; in original form 2023 February 2

ABSTRACT

The last decade has witnessed a rapid growth of the field of exoplanet discovery and characterization. However, several big challenges remain, many of which could be addressed using machine learning methodology. For instance, the most prolific method for detecting exoplanets and inferring several of their characteristics, transit photometry, is very sensitive to the presence of stellar spots. The current practice in the literature is identifying the effects of spots visually and correcting them manually or discarding the affected data. This paper explores a first step towards fully automating the efficient and precise derivation of transit depths from transit light curves in the presence of stellar spots. The primary focus of the paper is to present in detail a diverse arsenal of methods for doing so. The methods and results we present were obtained in the context of the 1st Machine Learning Challenge organized for the European Space Agency's upcoming *Ariel* mission. We first present the problem, the simulated *Ariel*-like data and outline the Challenge while identifying best practices for organizing similar challenges in the future. Finally, we present the solutions obtained by the top five winning teams, provide their code, and discuss their implications. Successful solutions either construct highly non-linear (w.r.t. the raw data) models with minimal pre-processing – deep neural networks and ensemble methods – or amount to obtaining meaningful statistics from the light curves, constructing linear models on which yields comparably good predictive performance.

Key words: Machine Learning – Data Methods – Exoplanets – Transit Photometry – Light Curves – Stellar Spots.

1. INTRODUCTION

In the coming decade, exoplanet atmospheric spectroscopy will undergo a revolution with a number of upcoming space and ground-based instruments providing unprecedented amounts of high-quality data. Most notable are of course the *Extremely Large Telescopes* (e.g. Gilmozzi & Spyromilio 2007; Nelson & Sanders 2008; Johns et al. 2012) on the ground and the *JWST* (Gardner et al. 2006) and the *Ariel* space telescope (Tinetti et al. 2016a). One of the outstanding challenges to high-precision spectrophotometry of exoplanets is the presence of stellar noise. Here, we will address in particular the presence of occulted star-spots in the spectrophotometric light curves of the *Ariel* space mission. The chromatic dependence of spots and faculae can adversely affect the measured exoplanetary transmission spectrum (through a biasing of the derived transit depth) as well as affect other light-curve parameters, such as limb-darkening and the mid-transit times. This is discussed in detail in references (e.g. Rabus et al. 2009; Nikolov et al. 2013; McCullough et al. 2014; Sing et al. 2015; Zellem et al. 2017; Rackham, Apai & Giampapa 2018, 2019;

★ E-mail: n.nikolaou@ucl.ac.uk

Iyer & Line 2020, and references therein). There exists a large body of literature on modelling star-spot signatures in photometric and radial velocity data (e.g. Dumusque et al. 2011; Lanza et al. 2011; Aigrain, Pont & Zucker 2012; Boisse, Bonfils & Santos 2012; Dumusque, Boisse & Santos 2014; Herrero et al. 2016; Gilbertson et al. 2020; Lisogorskyi et al. 2020; Zhao & Tinney 2020). Recently, Rosich et al. (2020) proposed a correction of the chromatic effects using Bayesian inverse modelling of long duration spectrophotometric time-series data with promising results. Carter et al. (2008) and Morris et al. (2018) propose detecting star-spots on transit light curves using methods based on Fisher information and the ratio of the ingress duration to total transit duration, respectively. To our knowledge, there exists no method designed to directly correct light curves of transits with crossing stellar spots.

In this publication, we explore the use of machine learning (ML) techniques to detect and correct for spot crossings in simulated data of the *Ariel* space mission. In particular, we report on the top five results of the 1st *Ariel* Mission ML Challenge (henceforth: the Challenge), which was concerned with the task of *correcting transiting exoplanet light curves for the presence of stellar spots*. The primary goal of the Challenge was thus to investigate if ML approaches are in principle suited to correcting star-spot crossings in spectrophotometric light curves across a large range of stellar and planetary parameters as well as observational signal-to-noise regimes.

To date, the use of ML approaches in exoplanets is still nascent but a burgeoning interest has seen the successful application of ML – and deep learning, in particular – to a variety of exoplanetary problems. These include (but are not limited to) the detection of exoplanet transits in survey data (e.g. Pearson, Palafox & Griffith 2018; Shallue & Vanderburg 2018; Osborn et al. 2020), the predictive modelling of planetary parameters (Lam & Kipping 2018; Alibert & Venturini 2019), instrument de-trending (e.g. Gibson et al. 2012; Waldmann 2012; Morello et al. 2014; Morvan et al. 2020), and the modelling and retrieval of atmospheric spectra (e.g. Waldmann 2016; Márquez-Neila et al. 2018; Zingales & Waldmann 2018; Cobb et al. 2019; Nixon & Madhusudhan 2020; Himes et al. 2022).

As with many problems in the field of exoplanetary science, the issue of star-spot crossings is characterized by a combination of challenges: (i) a large amount of data to process,[1] (ii) low signal-to-noise ratio, (iii) an underlying pattern which is non-linear and whose parametric form is a priori unknown, (iv) the available information comes in multiple forms (time dependent and independent), and finally (v) a high degree of degeneracy. These issues are commonly addressed by ML approaches.

This takes us to the second objective of the Challenge: promoting the interaction between the astrophysics and the ML communities. To this end, the Challenge targeted both audiences by being officially organized in the context of the ECML-PKDD 2019 conference[2] and also having a strong presence in the joint EPSC-DPS 2019[3] conference via a dedicated session. The Challenge ran from 2019 April to August. In total, 123 teams participated and it attracted the interest of researchers from both communities – as evidenced from the top five ranked teams and the solutions they submitted. As such, we consider the secondary objective of the Challenge has been met successfully. Building up on this success, a 2nd *Ariel* Data Challenge was organized in the context of ECML-PKDD 2021 and a 3rd one is currently under preparation.

But what of the main goal of the Challenge, i.e. automating the extraction of useful parameters from transiting exoplanet light curves in the presence of stellar spots? A large number of solutions outperformed our baselines and approached average precisions of 10 ppm in photometric flux for correctly predicting the relative transit depth per each wavelength from the noisy light curves. This was the case, despite exploring a generally high stellar spot coverage scenario (high activity stars), as discussed in Section 3.2. Again, building up on this success, the 2nd *Ariel* ML Challenge (2021) produced solutions that surpassed this performance (see Fig. 3), on a data set generated under a more realistic noise model covering full instrument systematics simulated under the ArielRad (Mugnai et al. 2020) and ExoSim (Sarkar et al. 2021) packages.

The solutions of the top five ranking teams that participated in the Challenge are presented in detail in this paper. Most solutions amount to constructing highly non-linear (w.r.t. the raw data) models with minimal pre-processing using deep neural networks (DNNs) and/or ensemble learning methods.[4] As we will see however, there exist comparably good – in terms of the precision of the obtained predictions – approaches that involve obtaining meaningful (i.e. informed by physics) statistics from the light curves and then training models that are linear w.r.t. them.

Just like the Challenge itself, this paper also intends to serve a dual purpose. Its primary goal is to describe the research problem of obtaining good predictions of the relative transit depth per each wavelength from simulated *Ariel*-like light curves distorted by photon noise and stellar spot noise, along with the solutions provided by the Challenge's winners and their implications. More specifically, the objective is to cover in detail a broad and diverse set of methods to attack the problem. Its secondary aim is to promote interaction between exoplanetary scientists and ML researchers. As such it is written in a language accessible to both audiences and – we hope – it contains useful information for exoplanetary scientists wishing to organize their own ML challenge or to refine their knowledge of ML methods and use them in their own work.

2. EXOPLANET BACKGROUND

Due to the interdisciplinary nature of this article, we here provide a very brief high-level introduction to transmission spectroscopy of exoplanets. Readers familiar with the field can safely skip this section, for a more in-depth review to exoplanetary spectroscopy we refer the reader to the relevant literature (e.g. Sharp & Burrows 2007; Tinetti et al. 2012; Madhusudhan 2019).

When a planet orbits its host star in our line of sight, we will observe a regular dimming of the stellar flux when the planet passes between us and the host star. This is referred to as a transit event. Similarly, when the planet is eclipsed by the host star, we will observe a small dip due to the loss of the planet's thermal or reflected light. In Fig. 1a, we show a schematic view of a transit and the resulting dip in the stellar flux time series, also known as a 'light curve'. The depth of this light curve, D, is typically of the order of 1 per cent for a Jupiter-sized planet and a Sun-like star. To first order, this dip can

[1] As *Ariel* is an upcoming space mission, the data in our case are obtained via simulations.

[2] ECML-PKDD, the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, is one of the leading academic conferences on ML and knowledge discovery, held in Europe every year.

[3] The European Planetary Science Congress (EPSC) and the American Astronomical Society's Division of Planetary Science (DPS) held a Joint Meeting at 2019.

[4] Ensemble methods are ML algorithms that construct powerful predictive models by combining multiple weaker predictors (Polikar 2006).

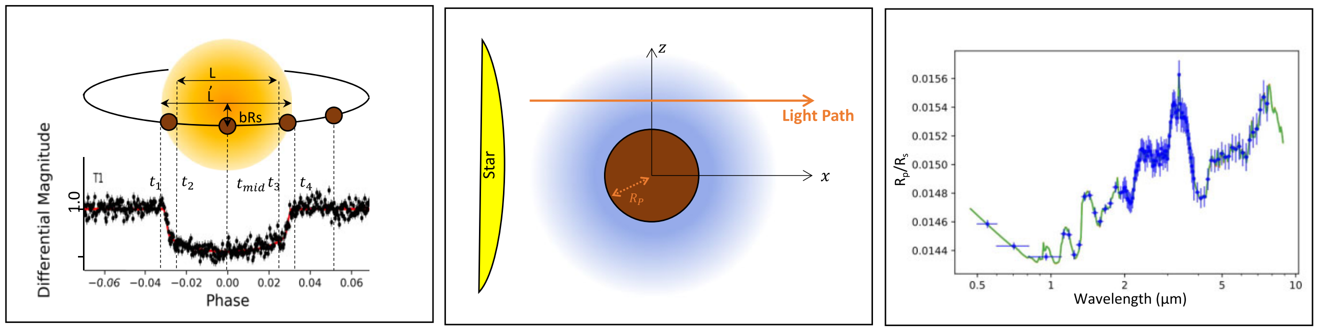


Figure 1. Left panel: Schematic of an exoplanet transit. The planet passes in-front of the star, obscuring some of the star's light. This leads to a characteristic dip in stellar flux observed as a function of orbital phase. Middle panel: Schematic view of transmission spectroscopy whereby some of the stellar light 'shines through' the gaseous envelope of a planet. Right panel: A simulated transmission spectrum of the *Ariel* mission. Blue points with errorbars correspond to the observed data points and green continuous curve corresponds to a theoretical atmospheric model. Figures courtesy of C. Changeat and adapted from Changeat et al. (2020).

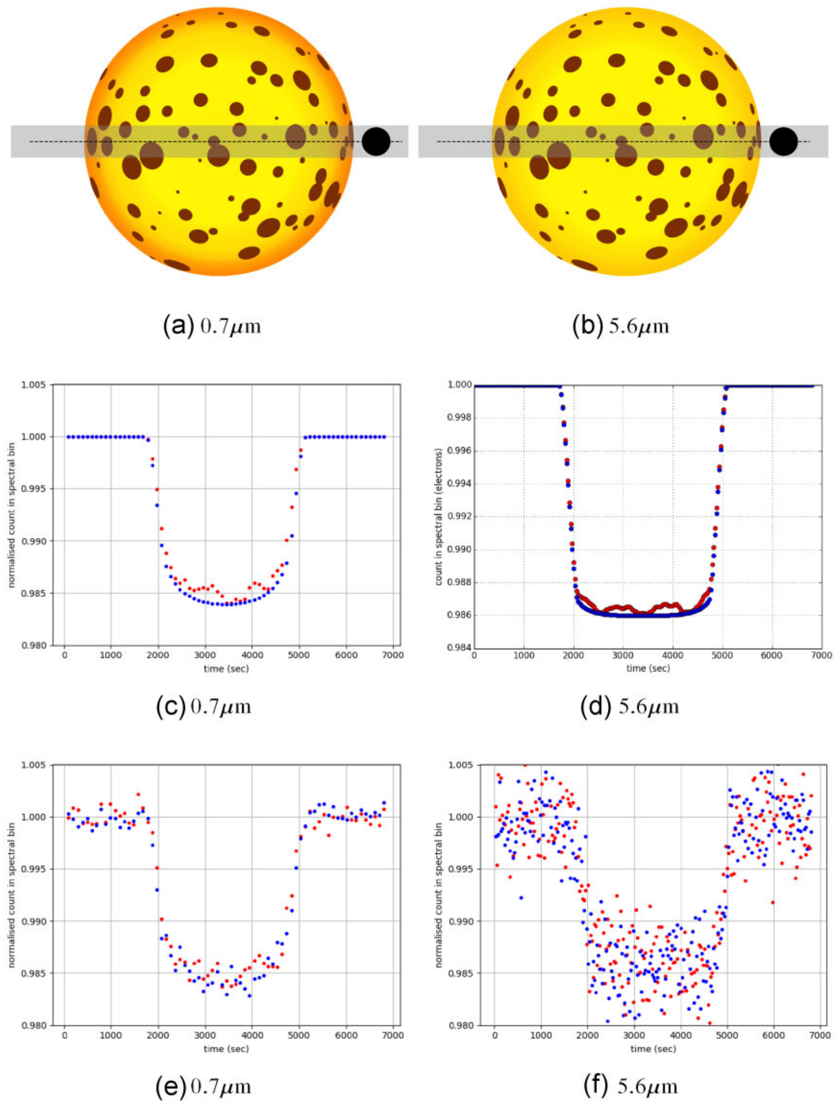


Figure 2. Examples of simulations including both stellar spots and faculae for 2 of the 55 wavelength channels, 0.7 and 5.6 μm. (a) and (b) Stellar surface simulations of a spotty star. Grey line shows the planet transit trajectory. The stellar surface limb brightness varies with wavelength. (c) and (d) Normalized observed flux as the planet transits across the star without stellar photon noise. Blue shows the perfect transit across a spotless star; red shows the transit across a spotty star. (e) and (f) Same as (c) and (d) but with stellar photon noise added.

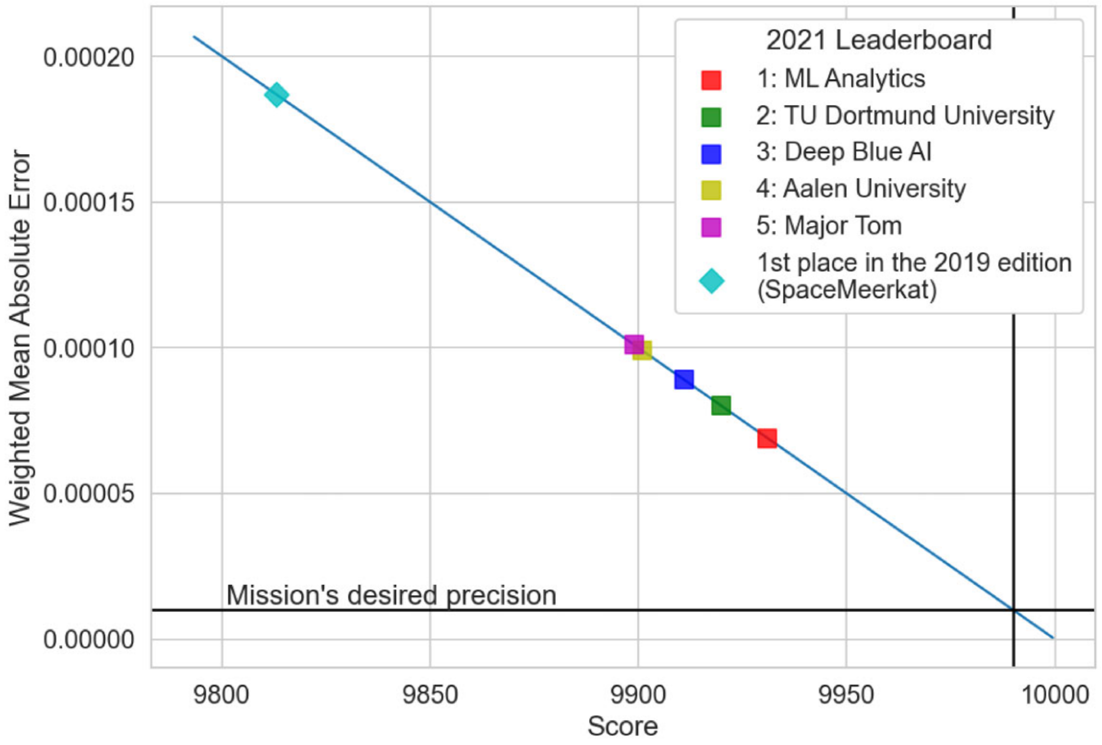


Figure 3. Progress made towards the *Ariel* mission's desired precision in terms of transit depth (10 ppm) by the 1st and 2nd *Ariel* ML Challenges. The winning solution of the 2021 Challenge (red) has a 61 per cent lower weighted MAE compared with the winning solution of the 2019 Challenge (teal). The solutions of both Challenges achieve a weighted MAE of the order of 10^{-5}–10^{-4}, despite exploring a high stellar spot coverage scenario (high activity stars). Note that results across the two Challenges are not directly comparable; in the 2019 Challenge (the focus of this work) instrument systematics were ignored, whereas in the 2021 Challenge, full instrument systematics were taken into account during the data set simulation, resulting in a more challenging modelling problem.

be described by the ratio of the planet to stellar radius, $D = R_p/R_*$ (also referred to as 'relative radius'). For an in depth explanation of the transit geometry, see Seager & Mallén-Ornelas (2003).

When a planet harbours an atmosphere, some of the stellar light will 'shine through' the planet's gaseous envelope (Fig. 1b). Depending on the atmospheric composition some light will be absorbed and/or scattered at specific wavelengths of light by the atmospheric gases, clouds, and aerosols. This leads to a wavelength-dependent 'loss' of stellar flux observed, which is equivalent to a perceived increase in planetary radius from an observational viewpoint. An accuracy of ca. 1 in 10^4 in flux measurements is typically required for a Jupiter-sized planet to observe this effect. Fig. 1c is a simulation of the resulting transmission spectrum of a hot Jupiter planet as observed by the *Ariel* space mission. The transmission spectrum includes absorption signatures of H_2O, CH_4, and CO as well as Rayleigh scattering and collision-induced absorption by hydrogen and helium (Changeat et al. 2020, fig. 1).

3. THE CHALLENGE

3.1 Data generation

For the purposes of the Challenge, we used the *Ariel* target-list produced by Edwards et al. (2019) to generate simulated light curves for all the 2097 planets in the list. For every planet, we produced 55 light curves, one for each wavelength channel corresponding to *Ariel* Tier 2 resolution (between 0.5 and 8.0 μm). In addition, all the light curves covered observations of 5 h, centred around the transit, with a time-step of 1 min. In reality, wavelength binning and time resolution will differ across targets of Tiers 1, 2, and 3 of *Ariel* Edwards et al. (2019). But here we opted to treat all targets as Tier 2 to make the data set accessible to all the participants without the need of renormalization (which would require knowledge on the transit modelling).

The simulated light curves were computed as follows:

(i) As a first step we calculated the limb-darkening coefficients (using the quadratic law) for every host star in the target list and for every wavelength channel. We used the EXOTETHYS package (Morello et al. 2020) and the stellar parameters for temperature and gravity provided in the target list, assuming zero metallicity for all the stars (the effect of metallicity is not strong). Also, we did not use the *Ariel* throughput as in this study we were only interested in the narrow wavelength channels, and any intra-channel variations due to the *Ariel* throughput are minimal.

(ii) We then calculated the planet-to-star radius ratio, R_p/R_*, for every planet in the target list and for every wavelength channel. This calculation was made using the TauRex atmospheric retrieval framework for exoplanets (Waldmann et al. 2015) and the planet parameters for temperature, mass, and radius (all provided in the target list), assuming the presence of water vapour and methane in the atmosphere with abundances that varied uniformly at random from planet to planet between 0.001 per cent and 0.1 per cent. The values for the abundances were an arbitrary choice, as the scope of using a spectrum was only to include some variability, of any kind, in the R_p/R_* parameter from one wavelength channel to another.

(iii) The next step was to define the spot model parameters for every host star in the target list. These parameters were:

(a) Spot coverage: This parameter corresponds to the percentage of the stellar surface that is covered by spots. We set this parameter to 10 per cent for every host star in the target list. In reality this parameter decreases with stellar temperature and initially we incorporated this in the model. However, it became clear that in the more realistic case, the number of spots that influence the light curves is very small, leading to almost noise-free data. For this reason, we chose to use the fixed value of 10 per cent in order to have a stronger spot effect on our data. This choice resulted in a simulated data set with many more spot-crossing events than in a real data set, suitable for the purposes of the challenge. We further justify this choice and discuss its implications in Section 3.2.

(b) Spot temperature: This parameter corresponds to the effective temperature of the spots, which is naturally lower that the effective temperature of the star. We calculated this parameter for every host star in the target list as a function of its temperature (T_*, provided in the target list), as described in Sarkar (2017), adjusted from Andersen & Korhonen (2015):

$$T_{\rm spot} = T_* - (0.0001\,343 \times T_*^2 - 0.6849 \times T_* + 1180.0). \quad (1)$$

(c) Spot contrast: This corresponds to the contrast between the brightness of the stellar surface and the brightness of the spots. We calculated this parameter for every host star in the target list and for every wavelength channel by integrating the respective PHOENIX stellar models (Husser et al. 2013) within each wavelength channel and dividing them.

(iv) Following the definition of the spot model parameters we created a set of spots for every host star in the target list. The spots were generated one by one, until the 10 per cent surface coverage was reached, and it was given three parameters:

(a) Latitude – uniformly at random generated number between −85 and 85 degrees,[5]

(b) Longitude – uniformly at random generated number between 0 and 360 degrees, and

(c) Angular diameter – randomly generated using a lognormal distribution, as described in Sarkar (2017), based on Bogdan et al. (1988):

$$\frac{{\rm d}N}{{\rm d}A} = M_A \exp\left[-\frac{(\ln A - \ln\langle A\rangle)^2}{2\ln\sigma_A}\right], \quad (2)$$

where N is the number of spots, A is the area of the spots, M_A is the maximum of the distribution (adjusted to result in 10 per cent of total coverage), $\langle A\rangle = 0.62 \times 10^{-6} A_{1/2\odot}$ is the mean of the distribution, and $\sigma_A = 3.8 \times 10^{-6} A_{1/2\odot}$ is the standard deviation of the distribution.

(v) With the set of spots generated for each star in the target list, we used the KSINT package (Montalto et al. 2014) to generate the spot-distorted light curves for every planet in the target list and for every wavelength channel. The input parameters for each light curve were: the set of spots, (number, position, and dimensions of all the spots), the spot contrast parameter, the limb-darkening coefficients, the planet-to-star radius ratio, the stellar density (calculated from the stellar mass and radius provided in the target list), and the planet orbital parameters (period and inclination, provided in the target list) and a viewing angle to make sure that the transit happens at the middle of the observation.

(vi) The final step was to add Gaussian noise to the light curve. No additional instrument systematics was assumed, as we aimed for the challenge to focus on correcting for the noise resulted from the stellar spots. The standard deviation of the Gaussian noise added was calculated from the overall noise on the transit depth estimation provided in the target list. This noise value depends on the stellar magnitude, the stellar temperature, the wavelength channel, and the characteristics of the *Ariel* instrument. It is beyond the scope of this work to describe exactly how this level of noise is estimated. We refer the interested reader to Edwards et al. (2019) for a detailed description.

This process resulted in generating data for 2097 simulated observations, consisting of 55 light curves each (one per wavelength). We repeated the process 10 times with different instances of the spot set (step iv). This resulted in 20970 simulated observations consisting of 55 light curves each, distorted by stellar spots. Finally, for each instance of the spot set, 10 different instances of additive Gaussian photon noise were introduced (step v). This resulted in 209 700 simulated observations consisting of 55 light curves each, distorted by both stellar spot and photon noise. These 209 700 simulated observations formed the final data set of the Challenge. The different instances of the spot set were included to mimic multi-epoch observations, were the spot pattern is expected to change, while the different instances of additive Gaussian photon noise were included to mimic continuous observation were the spot pattern is not expected to change. Note that the two sources of noise (spots and Gaussian) are treated as independent. Fig. 2 shows examples of light curves generated through this process. Most of the generated light curves only contained a single transit event; however, a small number of them included planets with small enough orbital periods to allow for multiple transits.[6]

Naturally, these details were unknown to the participants and neither were they used in the baseline solution. The aim of the Challenge was to infer the relative radii, either by explicitly modelling and subtracting, or by learning to ignore the photon and/or the stellar noise (or both).

3.2 On the choice of the 10 per cent spot coverage

It should be noted here that population studies of exoplanetary atmospheres tend to focus on low-activity stars (for which a 10 per cent spot coverage is high). This is because no robust methodologies for removing noise from stellar spots exist at the moment. However, here we chose to simulate a 10 per cent spot coverage as an example of a "hard case" to be addressed by the methodologies presented.

As we shall see, the results produced by these methods are encouraging in terms of being able to deal with relatively active stars. However, we should also note that the true stellar activity distributions, intensities, and morphologies are likely to be more complex than those simulated in this work.

Although the Challenge was organized in the context of *Ariel*, the methods presented here can be useful tools to analyse data from future missions as well, one of the outcomes of this work is that there

[5]We did not constrain the latitude further, in order to produce all possible scenarios: cases with spot-crossing events, cases with unocculted spots only, and cases with both occulted and unocculted spots. This variety was created by the combined effect of the uniform latitude distribution, the distribution of sizes, and the limit on the total area covered by the spots. A more restricted latitude would cause spot-crossing events in most cases; therefore, the solutions would tend to ignore the effects of unocculted spots.

[6]In case the light curve contained multiple transits, one of them was centred.

is no longer need to preclude younger or more active stars from future studies. This, in turn, can allow for population studies of exoplanets based on a larger and more representative sample.

Finally, we also need to note that even with a 10 per cent spot coverage on the star (uniformly distributed); a very large fraction of the simulated transits did not suffer from spot-crossing events.

3.3 Data set description and problem statement

Each *data point* (also known as an *observation* or an *example* in ML terminology) consists of a set of 55 noisy light curves (one per wavelength, corresponding to *Ariel* Tier 2 target resolution). Each light curve is a time series of 300 time-steps corresponding to 5 h of observation by *Ariel*. We shall denote with $x_{ij}^{(t)}$ the relative flux at time-step $t \in [1, 2, \dots, 300]$ of the light curve at wavelength $j \in [1, 2, \dots, 55]$ of the *i*th example. By $\boldsymbol{x}_{ij} = [x_{ij}^{(1)}, x_{ij}^{(2)}, \dots, x_{ij}^{(300)}]^\top$ we denote the entire light curve at wavelength *j* of the *i*th example. Finally, with $\boldsymbol{X}_i = [\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \dots, \boldsymbol{x}_{i55}]$ we denote all 55 light curves of the *i*th example.

Along with the light curves, six additional stellar and planetary parameters (all knowable in advance) were provided: the orbital period, stellar temperature, stellar surface gravity, stellar radius, stellar mass, and stellar *K* magnitude. We shall denote these as z_{i1}, $z_{i2}, \dots, z_{i6}$, respectively, for the *i*th example. Finally, with $\boldsymbol{z}_i = [z_{i1}, z_{i2}, \dots, z_{i6}]$ we shall refer to all the additional parameters of the *i*th example collectively.[7]

The noisy light curves and the six additional stellar and planetary parameters all constitute the quantities known in advance that we can use to alleviate the problem of stellar spots. In ML terminology they are the *features (independent variables)* in our prediction task.

The goal is to construct a model that uses these to predict a set of 55 real values, the relative radii $[R_p/R_*]_{ij}$ (one per wavelength *j*, for any given data point *i*). In ML terminology this is a *multitarget regression task*. The relative radii to be predicted are the *targets (dependent variables)* of the multitarget regression problem. For convenience, we shall henceforth denote the relative radius at wavelength *j* of the *i*th example, $[R_p/R_*]_{ij}$, with y_{ij}. Finally, with $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \dots, y_{i55}]$ we shall refer to all the relative radii of the *i*th example collectively. Note the planet to host star relative radius R_p/R_* is directly connected to the transit depth of the light curve, as the latter is equal to $\left(\frac{R_p}{R_*}\right)^2$.

The value of the 55 targets is known only for the *training examples* (the *statistical sample*). The goal of the learning task is – ideally – to construct a model $f(\boldsymbol{X}, \boldsymbol{z}) = \hat{\boldsymbol{y}}$ such that $\mathbb{E}[L(\boldsymbol{y}, \hat{\boldsymbol{y}})]$ is minimized, where $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$ denotes some measure of difference between the predictions $\hat{\boldsymbol{y}}$ and their corresponding true values $\boldsymbol{y}$ and $\mathbb{E}$ denote expectation over the joint distribution of $\boldsymbol{X}, \boldsymbol{z}$, and $\boldsymbol{y}$, i.e. – in statistical terminology – the underlying *population* from which the sample is drawn.

Once models are trained, they are evaluated on a separate *test set*. The predictive performance of a model on a previously unseen test set (drawn from the same distribution as the training set), serves as a proxy for its performance in the population, the latter being intractable. The features of the test set examples $\{(\boldsymbol{X}_i, \boldsymbol{z}_i) | i \in Test\}$ were provided to the participants and they had to upload their model's predictions $\{\hat{\boldsymbol{y}}_i | i \in Test\}$ on them. The ground truth $\{\boldsymbol{y}_i | i \in Test\}$ for the test set examples was unknown to the participants in the duration of the Challenge. It was only used to produce a ranking score for their submitted solution, which we describe in the next section.

[7]The values of the extra parameters provided were the same as those used to produce the simulations and no associated uncertainties were used.

3.4 Evaluation

All data points generated for a uniformly random set of 1677 out of the 2097 of the total planets (i.e. about 80 per cent of the generated data points) were used as training data. All data points generated for the remaining 420 planets were used to form the test set (i.e. were only used for evaluation). That is, the training and test sets not only contained no data point in common, but they also contained no data point from the same planet in common.

After producing a model (i.e. a solution to the problem), the participants could upload the predictions of the model on the Challenge's website. Subsequently, this would assign a score on the model based on the quality of the predictions. The participants were ranked on a leaderboard on the basis of their best solution and the progress of each participant's solutions in time was tracked to inform them of the impact of each change they made on the resulting model's predictive performance. The leaderboard ranking determined the winners of the Challenge that would receive prizes (top two participants) and the top five participants whose solutions we will present in Section 4.

The score assigned to each solution was a weighted average of the absolute error per target (i.e. on the relative radii) across all test set examples *i* and all wavelengths *j*:

$$Score = 10^4 - \frac{\sum_{i \in Test} \sum_{j=1}^{55} w_{ij} 2 y_{ij} |\hat{y}_{ij} - y_{ij}|}{\sum_{i \in Test} \sum_{j=1}^{55} w_{ij}} 10^6, \tag{3}$$

where y_{ij} is the true relative radius and $\hat{y}_{ij}$ the predicted relative radius of the *j*th wavelength of the *i*th test set example and the corresponding weight w_{ij} is given by:

$$w_{ij} = \frac{1}{{\sigma_{ij}}^2 {\delta_{F_{ij}}}^2}, \tag{4}$$

with ${\sigma_{ij}}^2$ being the variance of the relative stellar flux caused by the observing instrument at the *j*th wavelength of the *i*th example and ${\delta_{F_{ij}}}^2$ the variation of the relative stellar flux caused by stellar spots in the *j*th wavelength of the *i*th example. The value of σ_{ij} is an estimation based on an *Ariel*-like instrument, given its current design, while $\delta_{F_{ij}}$ is calculated based on stellar flux F_{ij}^{star} and the spot flux F_{ij}^{spot} in the *j*th wavelength of the *i*th example:

$$\delta_{F_{ij}} = 0.1 \left(1 - \frac{F_{ij}^{\text{spot}}}{F_{ij}^{\text{star}}}\right). \tag{5}$$

As we see, both sources of noise (photon and stellar spot) are wavelength-dependent and target-dependent (they depend on the star, therefore are different for each data point).

The higher the score, the better the solution's ranking. The maximum achievable score is 10 000 (if $\hat{y}_{ij} = y_{ij}, \forall i, j$). The score is not lower-bounded (i.e. can be negative), but even naive 'reasonable' models (e.g. predicting the average target value for all test data points) would not produce scores below 4000.

The weights w_{ij} of each target were unknown to the participants.[8] A sensible strategy would thus be to try to predict all of them reasonably well, in other words, to train a model to minimize an unweighted loss $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i)$ like the mean squared error (MSE), $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) = (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2$, the mean absolute error (MAE), $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) = |\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i|$, or their relative error counterparts: $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) = \left(\frac{\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i}{\hat{\boldsymbol{y}}_i}\right)^2$ or $L(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) = \frac{|\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i|}{\hat{\boldsymbol{y}}_i}$,

[8]For transparency of the evaluation process, the w_{ij} coefficients of the test set examples, along with the ground truth (target values y_{ij}) became available after the end of the Challenge.

respectively. Indeed, this is the approach taken by the top five participants and in training the baseline model.

3.4.1 Comparison with current practice

To assess the usefulness of the proposed ML-based solutions, in this paper we have included a comparison with the standard (non-ML-based) approach for obtaining estimates of R_p/R_s in the literature. In particular, we obtain least-squares fits of transit models using the `PyLightCurve` package (Tsiaras et al. 2016) on the entire test set, treating the stellar and transit parameters (orbital inclination, period, semi-major axis, stellar surface temperature, and gravity) as known, with the aim of estimating R_p/R_s. Limb-darkening coefficients are computed using the EXOTETHYS package (Morello et al. 2020) from the stellar gravity and temperature while assuming zero metallicity. This way, we are using all information available to the competition's participants – and only that – on the test set examples. Of course, such a model would be optimal in the absence of spots, but in this case the estimates of R_p/R_s will be biased because of the presence of spots.

As we will see in Table 1 the transit model fitting solution using `PyLightCurve`, achieves a score of 9467 under the competition's score metric of equation (3), which corresponds to an MAE on the estimated (R_p/R_s) of $0.00\,664 \pm 0.00\,006$ on the test set. Factoring for the weights of equation (4), the average weighted MAE is 0.000 533. Note that all top five solutions described in Section 4 attain a score higher than 9467 (i.e. better), the baseline solution of the competition given in Section 3.7, however, does not. Interestingly, all of the solutions that outperform our baseline also outperform the transit model least-squares fit solution.

3.5 Rules, logistics, and organization

To allow for the broadest possible participation, the set of rules of the Challenge was the minimal possible. There was no restriction on the models, algorithms, or data pre-processing techniques, neither on the programming languages, environments, or tools used for their implementation. The participants were also free to use data augmentation techniques, pre-trained models or any prior domain knowledge not included in the provided data set. Finally, they were free to choose their own way of splitting the training data between training and validation sets.

The participants were limited to one submission every 24 h. This was a measure taken to limit traffic on our website and – most crucially – to prevent the extent to which the solutions would be overfitting to the test set. Indeed, although the test set contains previously unseen examples by the model and the participants could not have access to the ground truth itself, the presence of a leaderboard is effectively causing some *information leakage from the test set*. Simply put, just adapting the strategies to the ranking score signal, participants could increase their scores by effectively overfitting on the particular test set. Limiting the number of daily submissions alleviated this effect. In retrospect, an even stronger strategy to prevent this would have been to only use part of the test set to produce the leaderboard ranking score during the Challenge and only use the full test set to produce the final ranking after the Challenge closes. In future ML challenges we will adopt this evaluation scheme. For now we should keep in mind that small differences in the ranking scores of solutions presented in Section 4 are not necessarily indicative of true generalization (i.e. ability to predict well on new examples).

The participants were allowed to form teams, provided they participated in only one entry. The remaining rules handled how prizes would be split among teams, how ties would be handled, and ensuring that any winning entry would have to beat the baseline model.

3.6 Description of solutions

To facilitate comparisons among the solutions discussed in the paper and to demonstrate the typical steps of training and evaluating models using ML methodology, we split the description of the solutions into three parts: (i) pre-processing, (ii) model/architecture, and (iii) training/optimization.

The 'pre-processing' part will describe any transformation of the raw data (either in terms of features or of observations) before giving it as input to a learning algorithm. The 'model' part is concerned with the general *class of models* (i.e. their *parametric form*) which the learning algorithm is exploring (e.g. DNNs of a given *architecture*,[9] random forests of 10 trees of maximal depth 5, and linear models of the form $y = ax_1 + bx_2 + c$). Finally, the 'training' part is concerned with the specifics of the optimization of the *parameters* of the model (i.e. the weights of the neural network, the derivation of the decision trees, or the inference of the linear coefficients a, b, and c in the examples mentioned ahead). It covers the *hyperparameters* used in the learning/optimization algorithm, along with the loss function it minimizes and the final evaluation method.

Wherever necessary, we will clarify the purposes behind modelling choices or training methodologies in all solutions described. However, a detailed treatment of models like DNNs is beyond the scope of this paper. We direct the interested reader to Goodfellow, Bengio & Courville (2016) and Chollet (2017).

3.7 Baseline solution

As a baseline ML solution, we trained a fully connected (FC) DNN[10] on a sample of 5000 training examples selected uniformly at random. The neural network uses all 55 noisy light curves, $\boldsymbol{X}_i$ to predict the 55 relative radii directly. It does not make use of any of the additional stellar and planetary parameters z_i.

3.7.1 Pre-processing

The noisy light curves have undergone the following pre-processing steps:

(i) Each light curve was smoothed using a moving median of window 3 (i.e. each value replaced by the median of itself and its two adjacent values). This was done to remove flux values that are obvious outliers.

(ii) In any light curve, any value (relative flux) that was above 1 was clipped to 1. This was done because the maximal relative flux during transit is 1.

(iii) All values were normalized for the transit depths to lie roughly within the range [0,1]. Doing so allows for faster and more stable training of models like DNNs. The normalization was carried out per wavelength and was performed as follows:

[9]By the term 'architecture' we collectively refer to the number, type, and connectivity of the neurons comprising a neural network.
[10]FC DNNs are the earliest and most popular type of DNN architecture. They are also known as multilayer perceptrons or 'dense' neural networks.

Table 1. Final leaderboard showing rank, score under equation (3), and mean weighted absolute error in terms of relative radius achieved by each of the top five entries and the baseline on the test data for the 1st *Ariel* ML Challenge (2019). We have included the results of the standard practice of fitting least-squares transit models on the test set light curves using PyLightCurve, assuming the stellar and planetary parameters as known. Results of the 2nd *Ariel* ML Challenge (2021) are also shown for reference.

Team	Rank	Score	Weighted MAE	Difference w.r.t. 1st place (per cent)
	Top five solutions of 1st *Ariel* ML Competition (2019)			
SpaceMeerkat	1	9813	0.000 187	
Major Tom	2	9812	0.000 188	+1
BV Labs	3	9808	0.000 192	+3
IWF-KNOW	4	9805	0.000 195	+4
TU Dortmund University	5	9795	0.000 205	+10
PyLightCurve Least Squares	14	9467	0.000 533	+185
2019 Baseline	15	8726	0.001 274	+581
	Top five solutions of 2nd *Ariel* ML Competition (2021) – for reference			
ML Analytics	1	9931	0.000 069	
TU Dortmund University	2	9920	0.000 080	+16
Deep Blue AI	3	9911	0.000 089	+29
Aalen University	4	9901	0.000 099	+43
Major Tom	5	9899	0.000 101	+46
2021 Baseline	27	9617	0.000 383	+455

First, we computed the average transit depths per wavelength from the target values $\bar{y}_j$ on a sample of 10 000 random training examples. For every wavelength j, we then applied the transformation:

$$x_{ij}^{(t)} \leftarrow (x_{ij}^{(t)} - (1 - 2\bar{y}_j^2))/2\bar{y}_j^2.$$

This was done to have the maximal relative flux values at exactly 1 and the transit depths around 0, leveraging the fact that the transit depths of the light curves are the squares of the relative radii (targets).

3.7.2 *Model/architecture*

We used a FC DNN with five two-dimensional (2D) hidden layers, all of which consisted of 1024 units × 55 channels, the jth channel receiving as input the light curve $\boldsymbol{x}_{ij}$ for each example. After these, we added a flattening layer followed by a linear layer of 55 outputs, the jth output corresponding to the predicted relative radius $\hat{y}_{ij}$ of each example. All other activation functions were rectified linear units (ReLUs).

3.7.3 *Training/optimization*

No batch normalization (BN), regularization, or dropout was applied in the training of the baseline model. The 5000 observations used were split into 4020 training and 980 validation examples (i.e. approximately 80 per cent training and 20 per cent validation split) in such a way that the two sets contained no planets in common. The model was trained by minimizing the average MSE across all wavelengths using the Adam optimizer (Kingma & Ba 2014) with a learning rate of 10^{-4} decaying with a rate of 0.01 and a batch size of 128. All remaining hyperparameters were set to default `Keras` (Chollet et al. 2015) values. The model was trained for a maximum number of five epochs without early stopping.

4. TOP FIVE SOLUTIONS

By the end of the Challenge, 13 out of the 123 participating teams had beaten the score attained by the baseline solution we just presented. In this section, we will present the top five ranked solutions. Their relative ranking in the final leaderboard and scores they achieved under equation (3), along with the weighted MAE in terms of relative radius are shown in Table 1. The table includes the results of the non-ML-based standard practice of fitting least-squares transit models on the test set light curves using `PylightCurve`, assuming the stellar and planetary parameters as known. As we see, although this method outperforms the competition's baseline, it is outperformed by all solutions that score above the baseline, including – of course – the top five solutions presented here. The results suggest that ML approaches can indeed outperform the current standard practice in the field, if implemented correctly.

For reference, Table 1 and Fig. 3, which show the progress made towards the *Ariel* mission's desired precision in recovering the relative radius from light curves contaminated with stellar spots, also include the results of the 2nd *Ariel* ML Challenge. Note that results across the two Challenges are not directly comparable, as the data were generated under different assumptions: in the 2019 Challenge (the focus of this work) instrument systematics were ignored, whereas in the 2021 Challenge, full instrument systematics were taken into account during the data set simulation (Mugnai et al. 2020; Sarkar et al. 2021), resulting in a more challenging modelling problem. As we see the solutions achieve a weighted MAE of the order of 10^{-5}–10^{-4}, despite exploring a high stellar spot coverage scenario (high activity stars), as discussed in Section 3.2. The analysis of top solutions of the 2021 Challenge will be the focus of future work.

4.1 SpaceMeerkat's solution

SpaceMeerkat is comprised of James M. Dawson, an Astrophysics PhD student at Cardiff University. *SpaceMeerkat*'s solution is a 1D convolutional neural network (CNN), designed to retain architectural simplicity, while exploiting the power of GPU-accelerated ML. The largest gain in the model's predictive power came from the extensive testing of different prepossessing operations.

4.1.1 Pre-processing

The data was split into 80 per cent training and 20 per cent test sets. In order to remove outlier flux values in the raw light curves, an initial smoothing was conducted on each time series $\boldsymbol{x}_{ij}$. The mean flux value in each non-overlapping bin of width 5 was calculated in-place along each time series leaving each observation $\boldsymbol{X}_i$ as a smoothed multichannel array of dimensions 60×55. A normalization operation was performed on the training set prior to its use for training ML models. For each of the 55 wavelengths, the medians across all data points of the lowest 1 per cent of flux values in each light curve for a given wavelength were calculated. These 55 percentile medians (henceforth 'median offsets') are therefore equal to

$$\kappa_j = \text{med}\{\boldsymbol{P}_{1\,\text{per cent}}(\boldsymbol{x}_{ij}^{(t')})\}, \tag{6}$$

where $\boldsymbol{P}_{1\,\text{per cent}}(\boldsymbol{x}_{ij}^{(t')})$ denotes the 1st percentile of the set of all flux values $x_{ij}^{(t')}$, $t' \in \{1, 2, \dots, 60\}$ for a given data point i and wavelength j, and $med\{\,\cdot\,\}$ denotes median across all data points i. The light curves were then divided by 1 minus the median offsets and the resulting flux values were thus

$$x_{ij}^{(t')} \leftarrow x_{ij}^{(t')}/(1-\kappa_j).$$

This normalization allowed the data to lie roughly within the range [0,1] but with leniency for allowing the existence of extremely shallow or deep transits. Any remaining flux values above the normalization range were clipped to 1. This was done to encourage the model to focus on the lower flux valued regions where most of the transit-depth information lies. The pre-processing of light curves makes use of `Astropy`,[11] a community-developed Python package for Astronomy (Astropy Collaboration 2013, 2018).

4.1.2 Model/architecture

The model used in this solution is a CNN (LeCun et al. 1995).[12] The data are presented to the CNN as a 1D vector and 1D convolutions and pooling operations are applied in order to maintain a principled simplicity to the final solution. The architecture of the CNN is shown in Table 2. The model was built using `PyTorch 0.4.1` (Paszke et al. 2019), an open source ML framework for Python users. The output of layer 'Lc5' in Table 2 is concatenated with the additional stellar and planetary parameters: the orbital period, stellar surface gravity, stellar radius, stellar mass, and stellar K magnitude, i.e. $[z_{i1}, z_{i3}, \dots, z_{i6}]$ for each example, to form the 1D linear input for layer 'Lc6'. The additional parameters did not undergo any normalization and were presented to the network in their raw form.

4.1.3 Training/optimization

The CNN was trained for 75 epochs (i.e. was presented with the entire training set 75 times), on a single NVIDIA TITAN Xp GPU. The model was trained using batches of 256 examples. Rather than presenting the CNN with examples of dimensions 60×55 (as generated by the pre-processing step), each example was flattened into a single vector of length 3300. Initial investigation showed that 1D convolutions over the flattened inputs produced significantly better results than 2D convolutions over the 2D pre-processed inputs. The model was trained by minimizing the MSE loss (see Section 3.4) using the standard Adam optimizer and an initial learning rate of 1×10^{-3} decaying by 10 per cent the existing rate, every epoch. No early stopping was used, as we observed no increase of the validation error during training to indicate the presence of overfitting. No additional form of regularization (e.g. BN, dropout, or explicit regularization) was used in the training procedure. All remaining hyperparameters were set to default `PyTorch` values. The code for this solution is publicly available on GitHub.[13]

Table 2. The CNN architecture used in the solution by the SpaceMeerkat team (ranked 1st). The table follows the standard PyTorch format. The 1st column lists the name of each layer/operation, the 2nd column its type, and the 3rd the dimensions of its output tensors (hence inputs to the next layer). These follow the convention (batch size, number of channels, height, and width). The filter column shows the dimensions (height and width) of kernels used to perform the convolution and pooling operations. Layer 'Lc6' is notable as this is where the additional planetary parameters **z** are introduced into the network.

Name	Layer/operation	Dimensions	Filter
Input	None	(256,1,1,3300)	None
Conv1	1D convolution	(256,32,1,3300)	(1,3)
ReLU	ReLU	None	None
AP1	1D average pool	(256,32,1,1650)	(1,2)
Conv2	1D convolution	(256,64,1,1650)	(1,3)
ReLU	ReLU	None	None
AP1	1D average pool	(256,64,1,825)	(1,2)
Conv3	1D convolution	(256,128,1,825)	(1,3)
ReLU	ReLU	None	None
AP1	1D average pool	(256,128,1,275)	(1,2)
Lc1	Linear	(256,1,1, 35200)	None
ReLU	ReLU	None	None
Lc2	Linear	(256,1,1, 2048)	None
ReLU	ReLU	None	None
Lc3	Linear	(256,1,1, 1024)	None
ReLU	ReLU	None	None
Lc4	Linear	(256,1,1, 512)	None
ReLU	ReLU	None	None
Lc5	Linear	(256,1,1, 256)	None
ReLU	ReLU	None	None
Lc6	Linear	(256,1,1, 60)	None
Output	None	(256,1,1,55)	None

4.2 Major Tom's solution

Major Tom took second place on the ARIEL ML Challenge scoreboard. The team composed of ML researchers from the Data Science Research and Analytics group at the University of Tuebingen (Germany). The goal of the team's solution is to provide an easy to use ML tool, with minimal data pre-processing effort and a fast inference step. The result is a fully integrated deep learning solution whose final code is publicly available online.[14]

[11] http://www.astropy.org

[12] CNNs are designed to excel in tasks in which translational invariance is important, i.e. we are looking for particular patterns anywhere in the input data. As such, they are especially popular in image-based tasks. However, they are very successful even outside this setting, as they effectively reduce the number of trainable parameters of a neural network (compared with a feedforward DNN of the same depth). This means they are more computationally efficient to train and more resistant to overfitting.

[13] Solution by *SpaceMeerkat* (ranked 1st): https://github.com/SpaceMeerkat/ARIEL-ML-Challenge.

[14] Solution by *Major Tom* (ranked 2nd): https://github.com/unnir/Ariel-Space-Mission-Machine-Learning-Challenge.

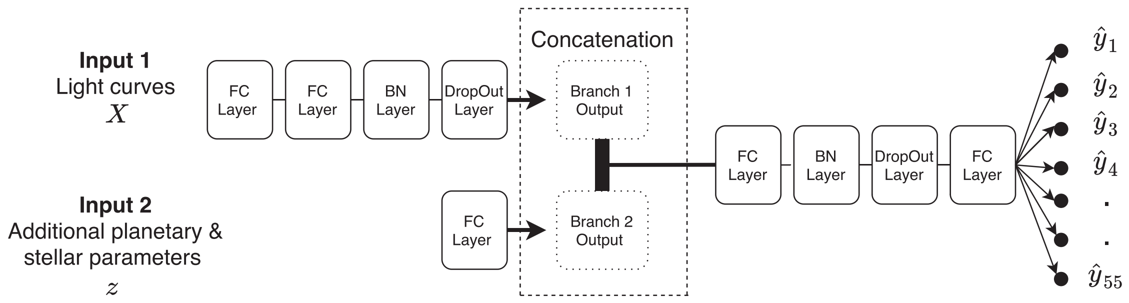


Figure 4. The deep learning model architecture proposed by the Major Tom team (ranked 2nd). The model has two separate inputs: one for the measurements $\boldsymbol{X}_i$, the second for the additional stellar and planetary parameters $\boldsymbol{z}_i$. The two branches are subsequently concatenated and higher level non-linear features combining information from both are extracted.

4.2.1 Pre-processing

The main motivation behind this solution was to create a robust statistical model that can handle outliers and noisy data. Therefore, we deliberately do not apply any heavy pre-processing to the data beyond the rescaling of the features and the targets. Since all measurements in time series $\boldsymbol{x}_{ij}$ are mostly distributed around 1 (see e.g. Fig. 5), we used the following rescaling of the data, in order to emphasize the differences between measurements:

$$x_{ij}^{(t)} \leftarrow (x_{ij}^{(t)} - 1) \times 1000.$$

We apply a similar transformation technique to the target variable y:

$$y_j \leftarrow y_j \times 1000.$$

4.2.2 Model/architecture

We used a multiple-input and multiple-output DNN model with FC, BN (Ioffe & Szegedy 2015), and dropout (Srivastava et al. 2014) layers.[15] The final architecture is presented in Fig. 4. It consists of two separate branches. The first branch uses as input the light curves $\boldsymbol{X}_i$, and the second, the additional stellar and planetary parameters $\boldsymbol{z}_i$. After several non-linear transformations, the outputs of the two branches are concatenated into one and higher level non-linear features combining information from both are extracted. The output layer has 55 neurons, the jth neuron mapping to the (rescaled) predicted relative radius y_{ij} of a given example. We utilized exponential linear unit activations in all but the last two layers, where ReLUs and linear activation functions are used, respectively.

4.2.3 Training/inference

We train the DNN using the *NAdam* optimization algorithm (Dozat 2016) and a *cyclic learning rate* as described in Smith (2017). The number of epochs was set to 1000, and the batch size to 3048. We selected the MSE as the loss function. We train the proposed model using 10-fold cross-validation with early stopping based on the validation loss with the patience equals to 20. The neural network was implemented using the KERAS/TENSORFLOW deep learning framework (Abadi et al. (2015)). The entire training step took ≈30 h using a single NVIDIA P100 GPU.

For the inference step, we used an ensemble consisting of all 10 models produced in the cross-validation steps; the final prediction is the average of all estimates from the 10 models.

[15]Both BN and dropout are commonly used techniques to prevent overfitting in neural networks.

4.3 BVLabs' solution

The team BVLabs took third place in the challenge. It is comprised of researchers and data scientists from the Jožef Stefan Institute and Bias Variance Labs. The team's solution relied on denoising the input data, the use of tree ensembles and FC neural networks.

4.3.1 Pre-processing

For each star–planet pair, we have 10 stellar spot noise instances and for each stellar spot noise instance we have 10 Gaussian noise instances. The data for each star–planet pair can therefore be represented as a tensor with dimensions (10, 10, 55, 300). For a fixed stellar spot noise instance, we computed the element-wise mean flux matrix over the 10 Gaussian noise instances which decreases the noise in the data. This can be seen as aggregating multiple measurements of the same target to decrease the variance of the observation. We are left with tensors with dimensions (10, 55, 300). Next, we compute element-wise medians over the 10 stellar spot noise instances, leaving us with tensors with dimensions (55, 300). An example of the result of this denoising process is presented in Fig. 5a.

The maximum flux (without noise) is always 1, whereas the minimal flux gives information about the planet radius. To further compensate for the noise, we do not use the minimal flux directly. Instead, we calculate two values: the minimum of the average of three consecutive flux values, and the median of the 10 lowest flux values. An example of the extracted values is shown in Fig. 5b.

We also estimated the amount of energy that stars emit at operating wavelengths of the ARIEL spacecraft. Tinetti et al. (2016b) list the five operating ranges of ARIEL. We divided each range into 11 bins of equal length, to get the estimates of the 55 wavelengths. To calculate the energy at a given wavelength, we used Planck's law

$$B(\lambda, T) \propto \frac{1}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda k_B T}\right) - 1},$$

where λ is the wavelength, h is the Planck's constant, k_B is the Boltzmann's constant, and c is the speed of light. The star temperature T was one of the six stellar and planetary parameters (see Section 3.1). In total, we used 171 ($3 \cdot 55 + 6$) features: three features for each of the 55 channels (the two extracted from the flux values and the energy emitted) and the six stellar and planetary parameters.

4.3.2 Model and training

Our best performing model was a heterogeneous ensemble consisting of three models. The first model was a random forest of 500 trees

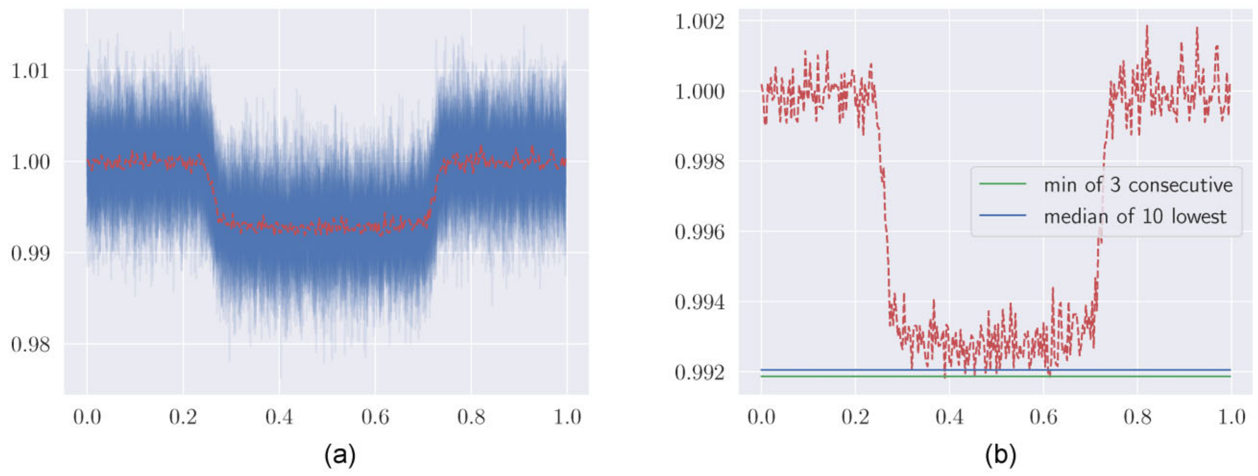


Figure 5. Pre-processing of the light curves by the BV Labs Team (ranked 3rd). Image (a) shows the light curve before (blue) and after (red) noise instance aggregation. Image (b) shows the features extracted from the denoised data. Both images show the data for star–planet pair 113, channel 25.

(Breiman 2001), as implemented in `scikit learn` (Pedregosa et al. 2011). The second model was an extreme gradient boosting (Friedman 2000) ensemble of 150 trees, as implemented in the `xgboost` library (Chen & Guestrin 2016). For both methods the parameters were optimized with cross-validation, and a separate model was learned for each channel. The third model was a multitarget (one model for all 55 channels) FC neural network with one hidden layer of 100 neurons. We used BN, dropout (with a rate of 0.2), and ReLU activations. The network was optimized with the Adam optimizer for 1000 epochs, with a constant learning rate 10^{-3}. As the loss function, average MSE across all targets was used. The network was implemented in `PyTorch`.

The weights of these three models in the final heterogeneous ensemble were optimized manually, with the best results obtained with a weight 0.15 assigned to the random forest, 0.25 to XGBoost, and 0.6 to the neural network. The code is available online.[16]

4.4 IWF-KNOW's solution

IWF-KNOW took the fourth place on the ARIEL ML Challenge scoreboard, and comprised of researchers and data scientists from the *Space Research Institute* (Austria), *Know-Center* (Austria), and the *University of Passau* (Germany). In contrast to the other top scorers who relied on deep learning approaches, their solution is based on a set of linear regressors, each of which is fast to train and easy to interpret (see Fig. 6). The corresponding scripts can be found on Zenodo.[17]

4.4.1 *Pre-processing*

We re-indexed the examples $\boldsymbol{X}_i$, each of size 300 × 55, in a new matrix $\boldsymbol{X}_{p,\,k,\,l}$, where $p \in \{1, 2, \ldots, 2097\}$ indexes the planet, $k \in \{1, 2, \ldots, 10\}$ the stellar spot instance, and $l \in \{1, 2, \ldots, 10\}$ the photon noise instance. To reduce the photon noise, we averaged the examples $\boldsymbol{X}_{p,\,k,\,l}$ over the photon noise instances l belonging to the same planet p and stellar spot noise instance k, yielding the noise-reduced example matrix $\widetilde{\boldsymbol{X}}_{p,k} = \frac{1}{10}\sum_{l=1}^{10}\boldsymbol{X}_{p,k,l}$. $\widetilde{\boldsymbol{X}}_{p,k}$ was of size 300 × 55 and comprised of the light curves for each wavelength.

Subsequently, we calculated the differences between the maxima and minima of each light curve in $\widetilde{\boldsymbol{X}}_{p,k}$. The maxima were assumed to be 1 as the light curves were already normalized, and the minima were estimated as the 1st, 5th, and 10th percentiles. This yielded estimates $\Delta F_{p,\,k,\,j,\,r}$ of the dip of the relative light curve caused by a transit of planet p for stellar spot noise instance k, wavelength j, and $r \in$ {1 per cent, 5 per cent, 10 per cent} corresponding to the 1st, 5th, and 10th percentiles. As the true dip $\Delta F_{p,\,j}$ of the relative light curve is approximately equal to the quadratic relative planet radius $\left(\frac{R_{p,j}}{R_{*,j}}\right)^2$, we took the square root of $\Delta F_{p,\,k,\,j,\,r}$ to obtain estimates of the relative planet radii:

$$\frac{R_{p,j}}{R_{*,j}} \approx \sqrt{\Delta F_{p,k,j,r}}.$$

We then built a feature vector $\boldsymbol{f}_p$ comprised of the estimated relative planet radii belonging to planet p:

$$\mathbf{f}_p = \left[\sqrt{\Delta F_{p,1,1,1\text{ per cent}}}, \ldots, \sqrt{\Delta F_{p,k,j,r}}, \ldots, \sqrt{\Delta F_{p,10,55,10\text{ per cent}}}\right].$$

The feature vectors $\boldsymbol{f}_p$ were augmented by the stellar and planetary parameters provided. For that, we averaged the six stellar and planetary parameters $z_{i1}, z_{i2}, \ldots, z_{i6}$ over all photon noise and stellar spot noise instances belonging to the same planet yielding $\boldsymbol{z}_p = [z_{p,\,1}, z_{p,\,2}, \ldots, z_{p,\,6}]$. The averaged stellar and planetary parameters $\boldsymbol{z}_p$ were then appended to the feature vectors $\boldsymbol{f}_p$ yielding the augmented feature vectors $\boldsymbol{f}^*_p$. The length of $\boldsymbol{f}^*_p$ was 1656, which resulted from 55 wavelengths, 3 percentile-based dip estimations, 10 spot noise instances, and 6 stellar and planetary features (55 × 3 × 10 + 6). Strictly speaking, the averaging was not necessary as the stellar and planetary parameters were the same for all instances of a planet (i.e. no noise was added to the stellar and planetary parameters). Finally, the extended feature vectors $\boldsymbol{f}^*_p$ were z-score normalized,[18] separately for the training and test set, thus avoiding information leakage from the test set into the training set.

We also re-indexed the scalar targets $y_{i,\,j}$ in the training set as $y_{p,\,k,\,l,\,j}$. Subsequently, we aggregated targets by averaging over all stellar spot noise instances k and photon noise instances l belonging to the same planet p, yielding the targets $y_{p,\,j}$. However, averaging

[16]Solution by *BV Labs* (ranked 3rd): https://github.com/bvl-ariel/bvl-ariel.
[17]Solution by *KNOW-IWF* (ranked 4rd) available under the DOI 10.5281/zenodo.3981141: https://doi.org/10.5281/zenodo.3981141.

[18]This type of normalization, also known as 'standardization' is performed by subtracting for each feature of a given example the mean value of that feature across all examples and dividing by its standard deviation.

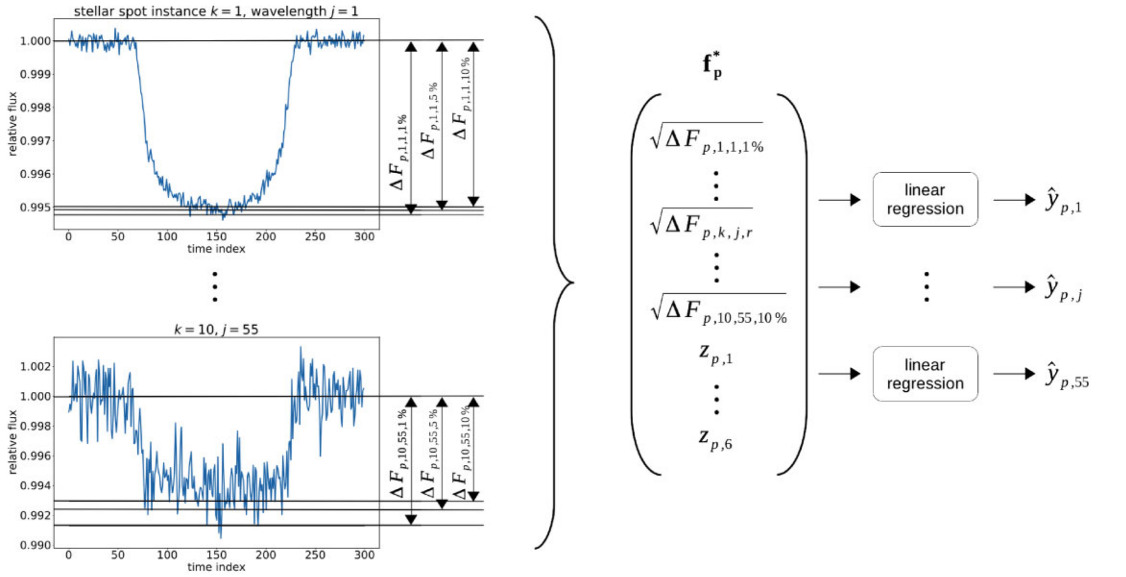


Figure 6. Regression pipeline of IWF-KNOW (ranked 4th). The light curves on the left are two examples in $\widetilde{X}_{p,k}$. The minima of the light curves were estimated using the 1st, 5th, and 10th percentiles. Subsequently, the minima were used to calculate the dips of the light curves $\Delta F_{p,k,j,r}$. The square root of all light curve dips $\Delta F_{p,k,j,r}$ belonging to the same planet p (i.e. including all wavelengths j and all stellar spot instances k), and additionally the stellar and planetary parameters $z_{p,1}, \dots, z_{p,6}$, were then gathered in the feature vector $\boldsymbol{f}_p^*$. The feature vector was z-score normalized (not shown in the graphic). Eventually, linear regressions were used to calculate the relative planet radius for each wavelength j.

was again not strictly necessary as all photon noise and stellar spot noise instances of a planet had the same relative radius in the provided data set.

4.4.2 *Model and training*

We set up a multiple linear regression model per wavelength j, resulting in 55 regression models:

$$y_{p,j} = \beta_{0,j} + \mathbf{f}_p^{*T} \boldsymbol{\beta}_j + \boldsymbol{\epsilon}_j$$

with $\boldsymbol{\beta}_j$ being the parameter vector of the model for wavelength j, $\beta_{0,j}$ the intercept term, and $\boldsymbol{\epsilon}_j$ the error term.

The parameters $\beta_{0,j}$ and $\boldsymbol{\beta}_j$ of the regression model were determined using least-squares estimation, which requires the estimation of the covariance matrix of $\boldsymbol{f}_p^*$. Because of the relatively large size of $\boldsymbol{f}_p^*$, we estimated the covariance matrix with the shrinkage method from Ledoit & Wolf (2004), which computes the shrinkage coefficient explicitly. The parameters were found using all examples from the training set. Following this, we used the regression models to predict all relative radii of the planets p in the test set with wavelength j:

$$\hat{y}_{p,j} = \beta_{0,j} + \mathbf{f}_p^{*T} \boldsymbol{\beta}_j .$$

The predicted relative radii $\hat{y}_{p,j}$ were re-indexed to the original indices $\hat{y}_{i,j}$ by copying $\hat{y}_{p,j}$ to all corresponding stellar spot noise instances and photon noise instances.

The only hyperparameters in our model were the percentiles used for estimating the minima of the light-curve dips. We found these parameters by trial and error and refrained from fine tuning them further.

4.5 TU Dortmund University

The team from TU Dortmund University, consisting of researchers working on applying ML algorithms in astroparticle physics, landed the 5th place on the leaderboard, going under the alias 'Basel321' during the Challenge. Their implementation is publicly available.[19] It embraces three central ideas: (i) the pre-processing simplifies the input time series, yet retains much of their information in auxiliary features; (ii) the baseline architecture is largely retained, but consists of two input branches: one using as input these auxiliary features and the other using as inputs the stellar and planetary parameters; and (iii) a bagging ensemble is created, in which each member is trained on data that have undergone slightly altered pre-processing.

4.5.1 *Pre-processing*

Fig. 7 shows how the input data are simplified by the use of z-scaled piecewise aggregate approximations (PAA, Keogh & Pazzani 2000), of which the lost information is retained in the auxiliary features μ, σ, and $\bar{\epsilon}$. These features describe each time series on a global level, while the PAA output captures the local shape. Namely, the PAA output is simply the average flux value in each of n_{paa} equal-sized segments. The z-scaled PAA representation facilitates learning due to the decreased number of dimensions and due to the uniform scale in each dimension. These properties are particularly relevant in dense neural networks like the baseline solution, which can suffer from a large number of model parameters if the input dimension is large.

4.5.2 *Model architecture and training*

An FC DNN is trained on the extracted features and the planetary and stellar parameters. The architecture used, shown in Fig. 8, is similar to the baseline, but it includes one branch for the auxiliary features and one for the planetary and stellar parameters. Fig. 8 also lists the associated hyperparameters.

[19]Solution by *TU Dortmund University* (ranked 5th): https://bitbucket.org/zagazao/ecml-discovery-challenge.

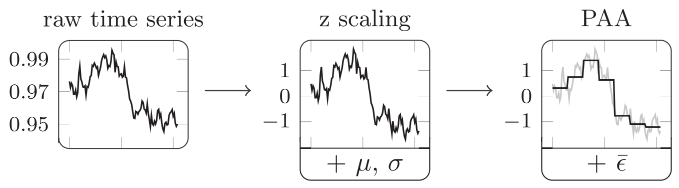
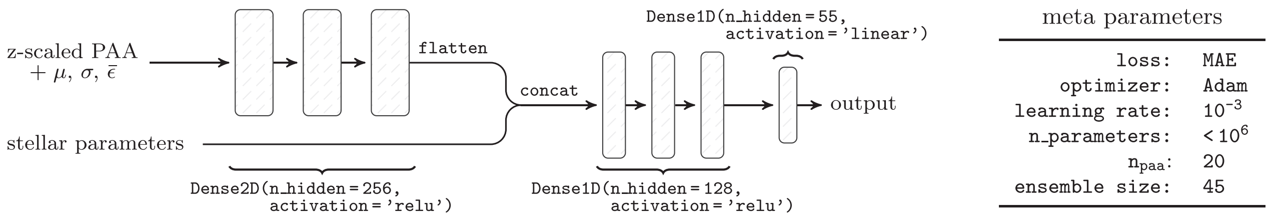


Figure 7. The TU Dortmund University team (ranked 5th) simplifies the raw data with PAAs of z-scaled time series. The information lost during these transformations is retained in auxiliary features. Namely, the z-scaling produces time series with zero mean and unit variance, but the original means $\mu \in \mathbb{R}$ and variances $\sigma \in \mathbb{R}$ of each channel and observation are kept. The PAA consists of only one average value in each equi-sized segment, but the overall reconstruction errors $\bar{\epsilon} \in \mathbb{R}$ are maintained.


z-scaled PAA
$+\ \mu,\ \sigma,\ \bar{\epsilon}$
stellar parameters
flatten
concat
output
Dense2D(n_hidden = 256, activation = 'relu')
Dense1D(n_hidden = 128, activation = 'relu')
Dense1D(n_hidden = 55, activation = 'linear')
meta parameters

loss:	MAE
optimizer:	Adam
learning rate:	10^{-3}
n_parameters:	$< 10^{6}$
$\mathtt{n}_{\mathrm{paa}}$:	20
ensemble size:	45

Figure 8. The first three layers of model used by the TU Dortmund University team (ranked 5th) derive abstract features from each time series that is represented by a PAA. The auxiliary features μ, σ, and $\bar{\epsilon}$ and the stellar parameters are also fed into the network. The last four layers combine these different kinds of inputs. A randomized parameter search has been employed to tune the number of layers and their size.

Multiple instances of the above architecture, were then combined in a bagging ensemble. To increase the diversity, each ensemble member shifted its input by a different number $n \in [0, n_{\mathrm{paa}})$ of time-steps. This alteration is performed already before the pre-processing, so that each ensemble member uses different PAA segments. The final prediction was the median among all ensemble members' predictions.

4.5.3 *Observations*

Regarding the feature representation extracted in the pre-processing step, we observed the following: (i) a linear regression on the z-scaled PAA representation is already able to outperform the baseline solution; (ii) it is critical to maintain the information lost during this type of pre-processing – this is achieved by the auxiliary features; and (iii) the use of shifting segments has remedied the fact that one set of PAA segments may not be optimal for all observations.

5. WHAT THE WINNING MODELS TEACH US

We should stress again that the final score differences among the top five ranked solutions, as shown on Table 1, are statistically negligible and should thus be regarded as equivalent in terms of predictive power in our simulated data. Having clarified this, these solutions provide us with some interesting insights with regards to the problem.

First of all, we observe that all five solutions make use of the additional stellar and planetary parameters (orbital period, stellar temperature, stellar surface gravity, stellar radius, stellar mass, and stellar K magnitude). This shows that these features indeed contain relevant information for uncovering the transit depths in light curves contaminated by the presence of stellar spots. Moreover, this information is not redundant given the noisy light curves.[20]

Another interesting observation is that most solutions involve the use of highly non-linear nonparametric or overparametrized[21] models w.r.t. the original features, like DNNs and/or ensembles of learners. More specifically, four out of five teams use deep learning approaches (*SpaceMeerkat*, *Major Tom*, *BV Labs*, and *TU Dortmund University*) and three out of five (*Major Tom*, *BV Labs*, and *TU Dortmund University* teams) use ensemble learning methods. The *Major Tom* team does not apply any pre-processing of the data provided beyond feature normalization, leaving all feature extraction to be implicitly performed by the DNN, using appropriate regularization techniques (BN and dropout) to prevent overfitting.

In contrast to this, the *IWF-KNOW* team relied on the extraction of non-linear features from the original inputs informed by domain knowledge. They then trained simple linear models in this new feature space.

The above are indicative of the non-linear nature of the problem. They also showcase the flexibility of ML and computational statistics methods in building models that capture this non-linearity. One can extract informative features given domain knowledge to capture it and then use simple and explainable models like linear regression trained on them. Alternatively, one can simply use powerful over-parametrized models, like DNNs and ensemble methods to implicitly learn transformations of the original feature space that are useful for the purposes of predicting the transit depth.

Extracting a small number of meaningful features informed by domain knowledge (*IWF-KNOW* and *BVLabs*) or appropriately

[20]It should be noted here, that the participants were given the 'exact' stellar and planetary parameters used to simulate the data. In reality, these are known with an associated degree of uncertainty which has not been taken into account here. We would therefore expect a degradation in terms of the performance of the algorithms to some degree, unless retrained on data accounting for these uncertainties.

[21]The term 'non-parametric' applies to models that are not restricted to a predetermined number of parameters. They can therefore adjust their complexity to the data at hand. Ensemble models can fall in this class. The term 'overparametrized' refers to parametric models having a number of learnable parameters that exceeds the number of data points. DNNs can fall in this class. Through appropriate use of regularization methods it is possible to avoid overfitting even when fitting models of such high complexity.

summarizing the light-curve information using signal processing techniques (*SpaceMeerkat* and *TU Dortmund University*) allows for simpler models to be trained in the lower dimensional extracted feature space. This allows for faster training and can also ultimately reduce overfitting.

A more detailed look into how the five solutions control for overfitting also reveals they follow quite different approaches. *SpaceMeerkat* uses a CNN rather than an FC DNN to reduce the number of effective learnable parameters. *Major Tom* uses an FC DNN but controls for its complexity via BN, dropout, and the use of an ensemble of trained DNNs, rather than a single model. *BV Labs* also make extensive use of ensembling and their neural network learner also uses BN and dropout. The fact that they operate on a much lower dimensional feature space (only 171 features per data point) also aids in reducing overfitting. *IWF-KNOW* use linear regression models, which are characterized with high bias (i.e. more prone to underfitting than overfitting). They also operate on a lower dimensional space (1656 features per data point) and apply shrinkage. Last but not least, *TU Dortmund University* makes use of an ensemble which is interestingly built on data having undergone slightly different pre-processing. Training on perturbed inputs results in making them more robust to overfitting.

Two of the top five teams (*BV Labs* and *IWF-KNOW*) made use of the fact that the training data contained multiple data points corresponding to the same planet under (10 different photon noise and 10 different stellar spot noise instances). They treated the two noise sources as independent and averaged these out or took the median to obtain less noisy light curves. This was a sensible thing to do and such a scenario would indeed occur if multiple observations of the same target were to be obtained.

Finally, ignoring outlier flux values via smoothing/down-sampling the light curves (*SpaceMeerkat*), clipping values above 1 (*SpaceMeerkat* and *BVLabs*), or by extracting summary statistics from the light curve and using them as features (*SpaceMeerkat*, *BVLabs*, *IWF-KNOW*, and *TU Dortmund University*) proved a useful strategy in building more robust models.

6. CONCLUSIONS

Correcting transit light curves for the effects of stellar spots is a challenging problem, progress in which can have a high impact on exoplanetary science and exoplanet atmosphere characterization in particular.

The primary goal of the *Ariel mission's 1st ML Challenge* was to investigate the existence of fully automated solutions to this task that predict the transit depth with a precision of the order of 10^{-5}–10^{-4} with the use of ML and computational statistics methodologies. The secondary goal was to bridge the ML and exoplanetary science communities. As we saw, both of these goals were met with success.

The aim of this work is to serve as a starting point for further interaction between the two communities. We described the data generation, the problem outline, and the organizational aspects of the Challenge. We intend this to serve as a reference for the organization of future challenges in data analysis for exoplanetary science. In the interests of communicating the modelling outcomes of the Challenge, we also presented, analysed, and compared the top five ranked solutions submitted by the participants.

As evidenced by the top five entries, the Challenge indeed attracted the interest of both exoplanetary scientists and ML experts. The participants cover an impressive breadth of academic backgrounds and the submitted solutions an equally impressive range of approaches, from linear regression to CNNs.

The solutions obtained demonstrate that it is indeed feasible to fully automate the process of efficiently correcting light curves for the effect of stellar spots to the desired precision. One key insight obtained is that additional stellar and planetary parameters (orbital period, stellar temperature, stellar surface gravity, stellar radius, stellar mass, and stellar *K* magnitude) can greatly improve the derivation of correct transit depths in the face of stellar spots. Moreover, although the scenarios examined where characterized by high spot coverage (10 per cent), the results suggest that it is possible to successfully correct even such ‘hard cases’. This, in turn, suggests that younger or more active stars need not be excluded from atmospheric population studies due to data analysis limitations. Planets orbiting such stars can thus be targeted by future space missions.

Good solutions can be obtained by a wide range of modelling methodologies. They include simple, easily interpretable models, like linear regression, built on features derived from clever feature engineering, informed by exoplanet science theory. Other solutions amount to training complex ML models using deep learning or ensemble learning, which automate the extraction of useful features from minimally pre-processed – even raw – data. In the latter case, especially for DNN models it is crucial to take measures to prevent overfitting. These can include dimensionality reduction, ensembling, use of convolutional filters, BN, dropout, training using perturbed data, and combinations thereof.

The next steps of this work include refinement of the proposed solutions to handle more realistic simulated data, possibly involving both stellar spots and faculae (areas of the host star characterized by increased temperature). Upon successful performance on these, the provided solutions can then be used in the analysis pipeline of *Ariel* data or adapted to other instruments.

SUPPORTING INFORMATION

Supplementary data are available at RASTAI online.

suppl_data

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.

ACKNOWLEDGEMENTS

NN acknowledges the support of the Nvidia Corporation through the NVIDIA GPU Grant program. JD has received funding by the Science and Technology Facilities Council (STFC) as part of the Cardiff, Swansea & Bristol Centre for Doctoral Training. He gratefully acknowledges the support of the Nvidia Corporation with the donation of a Titan Xp GPU used for this research and the help of Nikki Zabel for the completion of this work. TS and MP thank Martin Breskvar, Dragi Kocev, and Nikola Simidjievski for their contributions, and Barbara Krašovec for her help with the Slovenian national supercomputing network. RLB thanks the Austrian Science Fund (FWF) for funding: P31659-N27. The team from TU Dortmund University has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038A), and by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 ‘Providing Information by Resource-Constrained Data Analysis’, project C3. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 758892, ExoAI). Furthermore, we acknowledge funding by the Science and Technology Funding Council (STFC) grants: ST/K502406/1, ST/P000282/1, ST/P002153/1, and ST/S002634/1.

DATA AVAILABILITY

The data underlying this article are available in Zenodo, at https://doi.org/10.5281/zenodo.10043836.

REFERENCES

Abadi M. et al., 2015, preprint (arXiv:1603.04467)
Aigrain S., Pont F., Zucker S., 2012, MNRAS, 419, 3147
Alibert Y., Venturini J., 2019, A&A, 626, A21
Andersen J. M., Korhonen H., 2015, MNRAS, 448, 3053
Astropy Collaboration, 2013, A&A, 558, A33
Astropy Collaboration, 2018, AJ, 156, 123
Bogdan T. J., Gilman P. A., Lerche I., Howard R., 1988, ApJ, 327, 451
Boisse I., Bonfils X., Santos N. C., 2012, A&A, 545, A109
Breiman L., 2001, Mach. Learn., 45, 5
Carter J. A., Yee J. C., Eastman J., Gaudi B. S., Winn J. N., 2008, ApJ, 689, 499
Changeat Q., Keyte L., Waldmann I. P., Tinetti G., 2020, ApJ, 896, 107
Chen T., Guestrin C., 2016, Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16. ACM, New York, p. 785
Chollet F., 2017, Deep Learning with Python, 1st edn. Manning Publications Co., USA
Chollet F. et al., 2015, Keras. Available at: https://keras.io/getting_started/faq/#how-should-i-cite-keras
Cobb A. D. et al., 2019, AJ, 158, 33
Dozat T., 2016 ICLR Workshop, 1, 2013
Dumusque X., Santos N. C., Udry S., Lovis C., Bonfils X., 2011, A&A, 527, A82
Dumusque X., Boisse I., Santos N. C., 2014, ApJ, 796, 132
Edwards B., Mugnai L., Tinetti G., Pascale E., Sarkar S., 2019, AJ, 157, 242
Friedman J. H., 2000, Ann. Stat., 29, 1189
Gardner J. P. et al., 2006, Space Sci. Rev., 123, 485
Gibson N. P., Aigrain S., Roberts S., Evans T. M., Osborne M., Pont F., 2012, MNRAS, 419, 2683
Gilbertson C., Ford E. B., Jones D. E., Stenning D. C., 2020, ApJ, 950, 155
Gilmozzi R., Spyromilio J., 2007, The Messenger, 127, 3
Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge, MA
Herrero E., Ribas I., Jordi C., Morales J. C., Perger M., Rosich A., 2016, A&A, 586, A131
Himes M. D. et al., 2022, Planet. Sci. J., 3, 91
Husser T. O., Wende-von Berg S., Dreizler S., Homeier D., Reiners A., Barman T., Hauschildt P. H., 2013, A&A, 553, A6
Ioffe S., Szegedy C., 2015, *Proc. Machine Learning Research* , 37, 448
Iyer A. R., Line M. R., 2020, ApJ, 889, 78
Johns M. et al., 2012, Proc. SPIE Conf. Ser. Vol. 8444, Ground-based and Airborne Telescopes IV. SPIE, Bellingham, p. 84441H
Keogh E. J., Pazzani M. J., 2000, Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min., Scaling Up Dynamic Time Warping for Datamining Applications. ACM, New York, p. 285
Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
Lam C., Kipping D., 2018, MNRAS, 476, 5692
Lanza A. F., Boisse I., Bouchy F., Bonomo A. S., Moutou C., 2011, A&A, 533, A44
LeCun Y., Bengio Y., 1995, The Handbook of Brain Theory and Neural Networks, 3361, 1995
Ledoit O., Wolf M., 2004, J. Multivariate Anal., 88, 365
Lisogorskyi M. et al., 2020, MNRAS, 497, 4009
McCullough P. R., Crouzet N., Deming D., Madhusudhan N., 2014, ApJ, 791, 55
Madhusudhan N., 2019, ARA&A, 57, 617
Márquez-Neila P., Fisher C., Sznitman R., Heng K., 2018, Nature Astron., 2, 719
Montalto M., Boué G., Oshagh M., Boisse I., Bruno G., Santos N. C., 2014, MNRAS, 444, 1721
Morello G., Waldmann I. P., Tinetti G., Peres G., Micela G., Howarth I. D., 2014, ApJ, 786, 22
Morello G., Claret A., Martin-Lagarde M., Cossou C., Tsiaras A., Lagage P. O., 2020, AJ, 159, 75
Morris B. M., Agol E., Hebb L., Hawley S. L., 2018, AJ, 156, 91
Morvan M., Nikolaou N., Tsiaras A., Waldmann I. P., 2020, AJ, 159, 109
Mugnai L. V., Pascale E., Edwards B., Papageorgiou A., Sarkar S., 2020, Exp. Astron., 50, 303
Nelson J., Sanders G. H., 2008, Proc. SPIE Conf. Ser. Vol. 7012, Ground-based and Airborne Telescopes II. SPIE, Bellingham, p. 70121A
Nikolov N. et al., 2013, MNRAS, 437, 46
Nixon M. C., Madhusudhan N., 2020, MNRAS, 496, 269
Osborn H. P. et al., 2020, A&A, 633, A53
Paszke A. et al., 2019, Advances in Neural Information Processing Systems 32. Curran Associates, Inc., Canada, p. 8024
Pearson K. A., Palafox L., Griffith C. A., 2018, MNRAS, 474, 478
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Polikar R., 2006, IEEE Circ. Syst. Mag., 6, 21
Rabus M. et al., 2009, A&A, 494, 391
Rackham B. V., Apai D., Giampapa M. S., 2018, ApJ, 853, 122
Rackham B. V., Apai D., Giampapa M. S., 2019, AJ, 157, 96
Rosich A., Herrero E., Mallonn M., Ribas I., Morales J. C., Perger M., Anglada-Escudé G., Granzer T., 2020, A&A, 641, A82
Sarkar S., 2017, PhD thesis, Cardiff Univ.
Sarkar S., Pascale E., Papageorgiou A., Johnson L. J., Waldmann I., 2021, Exp. Astron., 51, 287
Seager S., Mallén-Ornelas G., 2003, ApJ, 585, 1038
Shallue C. J., Vanderburg A., 2018, AJ, 155, 94
Sharp C. M., Burrows A., 2007, ApJS, 168, 140
Sing D. K. et al., 2015, Nature, 529, 59
Smith L. N., 2017, 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, New York, p. 464
Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929
Tinetti G., Tennyson J., Griffith C. A., Waldmann I., 2012, Phil. Trans. R. Soc. A, 370, 2749
Tinetti G. et al., 2016a, Proc. SPIE Conf. Ser. Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 99041X
Tinetti G. et al., 2016b, Proc. SPIE Conf. Ser. Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, Bellingham, p. 658
Tsiaras A., Waldmann I., Rocchetto M., Varley R., Morello G., Damiano M., Tinetti G., 2016, ApJ, 832, 202
Waldmann I. P., 2012, ApJ, 747, 12
Waldmann I. P., 2016, ApJ, 820, 107
Waldmann I. P., Tinetti G., Rocchetto M., Barton E. J., Yurchenko S. N., Tennyson J., 2015, ApJ, 802, 107
Zellem R. T. et al., 2017, ApJ, 844, 27
Zhao J., Tinney C. G., 2020, MNRAS, 491, 4131
Zingales T., Waldmann I. P., 2018, AJ, 156, 268