



Full length article

Synthetic light curves of exoplanet transit using nanosatellite data

A. Fuentes^a, M. Solar^{b,*}^a Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso, Chile^b Universidad Técnica Federico Santa María, Av. Vicuña Mackenna 3939, Santiago, Chile

ARTICLE INFO

Keywords:

Deep learning
 Datasets
 Exoplanet detection
 Nanosatellite data
 Artificial intelligence

ABSTRACT

In this article, we present a dataset of light curves with synthetic signals. BRITE light curves (a constellation of five nanosatellites) are the main source of this dataset. We create the synthetic light curves of exoplanet transit by applying a pre-processing to the BRITE data and an injection of transit according to the Mandel and Agol model with a constraint of stellar radius $< 3.08[R_{\text{sun}}]$ and planetary radius between 0.95 and $2.1[R_{\text{jup}}]$. We apply a quality criterion, obtaining 597 Planet Candidate (PC) examples and 3126 Not Planet Candidate examples as a dataset. PCs are injected simulated planets and are not around unique stars. We design a Deep Learning (DL) model to be trained with the created dataset. The DL model is a modified AstroNet Convolutional Neural Network (CNN) from literature to detect possible exoplanets. After evaluation over the testing set we obtain an accuracy of 99.46%, precision of 100% ($PC_{\text{precision}}$) and a recall of 96.72% for the PC class (PC_{recall}), and an area under the curve receiver operating characteristics ($AUC - ROC$) of 100%, overcoming the results of existing networks tested on BRITE data. We ultimately search for potential exoplanets using the pre-processed data from BRITE, finding signals similar to exoplanetary transits in the targets HD 039060, HD 022049, HD 036861 and HD 218396.

1. Introduction

Before the space race, astronomy was limited to the study of electromagnetic radiation that can cross the atmosphere that separates us from outer space. With the advances in satellite technology, the doors were opened to the astronomical study of higher frequencies, and the study of less distorted light as it does not cross our dense atmosphere. In this way, various physical phenomena are observed through different wavelengths by multiple astronomical space missions to complement terrestrial observations, e.g. planets that orbit stars other than the Sun. Deeg and Belmonte (2018) show up-to-date references of exoplanets.

The main mission by the number of exoplanets discovered and confirmed (2778 exoplanets as of 26/01/2024) is the Kepler mission, designed to determine the frequency of exoplanets the size of Earth or larger in the habitable zone of stars similar to the Sun to characterize the diversity of planetary systems (Borucki, 2016; Murphy, 2012).

A pipeline is utilized to convert the data downloaded from the satellite into fluxes through calibration of the pixel data, performing aperture photometry and correcting systematic errors. The data generated is stored publicly in MAST (Mikulski Archive for Space Telescopes), and light curves can be downloaded from the host¹ as Long

Cadence (LC), Short Cadence (SC) and target pixel files. The extended mission, called K2, was prolific with 548 exoplanets confirmed as of 26/01/2024 (Vanderburg and Johnson, 2014).

Some exoplanets that have been discovered, like Earth, are not found alone. There are other exoplanets orbiting the same star. As of 26/01/2024, NASA (2024) has 5572 confirmed exoplanets using data from both space and terrestrial observatories.

Following Kepler and K2, the Transiting Exoplanet Survey Satellite (TESS) has the highest number of exoplanets discovered and to be confirmed (Ricker et al., 2016). The TESS mission has captured images of tens of millions of stars (Kunimoto et al., 2022), and there are currently 415 exoplanets confirmed and 7027 expected to be confirmed (26/01/2024). The public data of TESS, like Kepler, are located at MAST and NASA Exoplanet Archive.²

Analogous to Kepler, the TESS Input Catalog (TIC) is used to select mission objectives and calculate the physical and observational properties of the candidates for exoplanet. Its release 8 uses the GAIA DR2 catalog as a base and merges a large number of other photometric catalogs such as 2MASS, UCAC4, APASS, SDSS, WISE, etc. (Centre de Données astronomiques de Strasbourg, 2022). Also, there is the Candidate Target List (CTL), whose purpose is to provide a subset of

* Corresponding author.

E-mail address: mauricio.solar@usm.cl (M. Solar).¹ <https://archive.stsci.edu/>² <https://exoplanetarchive.ipac.caltech.edu/>

TIC objects to select the target stars for 2 min cadence observations in service of the primary scientific requirements of TESS (Stassun et al., 2018).

Despite the scientific benefits of mission study space, they have a significant disadvantage with respect to terrestrial observatories due to their high cost of designing, development, launch, and maintenance. To put in perspective, Kepler's life-cycle cost was approximately 600 million USD (Serjeant et al., 2020). In this context, the use of nanosatellites has been explored for scientific research in the areas of astronomy and astrophysics. For example, the CubeSat standard has gained popularity in space agencies and research centers due to its rapid development and low costs compared to large missions. The standard size of nanosatellites is built in units of $10 \times 10 \times 10$ [cm³] called 1U, maintaining a weight normally less than 2 [Kg]. The cost of a research-grade astrophysics CubeSat is between 5 and 10 million USD (Shkolnik, 2018).

Shkolnik (2018), Serjeant et al. (2020) and Douglas et al. (2019) present the state of the art of the possibilities of use of nanosatellites to study astronomical phenomena. They describe past, current and future missions specifying the objectives, technological opportunities and electromagnetic spectrum to be studied.

The Arcsecond Space Telescope Enabling Research In Astrophysics (ASTERIA) is a CubeSat (6U) nanosatellite designed to demonstrate high precision photometric technologies with a small platform (Knapp et al., 2020). ASTERIA's scientific objectives also involve studying the Sun-type star 55 Cancri, which has five exoplanets, including one (55 Cancri e) that transits every 18 h. The budget for ASTERIA mission was 8.2 million USD (Smith et al., 2018).

We are highlighting the BRITE mission, which consists of five nanosatellites that observe bright stars. The data from this mission is relevant since it may contain possible exoplanetary transits. BRITE light curves are the main source of data of this article, including downloadable curves for 609 different stars (Weiss et al., 2014). Due to the radiation problems, stare mode photometry was strongly affected. In response, chopping mode is the default mode used in BRITE for data collection to date. Popowicz et al. (2017) present the data processing and photometry.

Detecting exoplanets is a significant scientific and engineering achievement since they are difficult to detect due to their faint shine compared to the bright stars that orbit. Different detection methods are generally used to detect exoplanets (Lang, 2013). Some of them are direct imaging, astrometry, radial velocity, transit event observation, microlensing (Dai et al., 2021) and transit-timing variation (Wright and Gaudi, 2012).

Among these techniques is the transit method which consists of studying variations of light from a star over time to find a characteristic signal that indicates a planet passing between the star and an observing telescope, an event that is periodic (Wright and Gaudi, 2012).

A relevant aspect is the period of the event, the time in which it happens and the characteristics of the planet's orbit. Even knowing the period, it is necessary to observe it for long periods of time to verify if it really is a periodic signal.

The transit technique detects a possible exoplanet as it considers that its presence accompanying a star generates temporal variations in the flux of the system with respect to a system without the exoplanet. This temporal variation is detectable through periodic attenuations of the detected signal. In this context, data are studied from sequential observations over time, known as Time Series. These series known as light curves consist of flux or magnitude values of an astronomical object over a certain period of regular or irregular time.

The search for hidden periods in time series data is studied with Spectral Analysis, which provides tools to discover periodicities of signals from different phenomena such as acoustic, communication, biomedical, science, etc. (Cryer and Chan, 2008). According to the characteristics of the data, there are different techniques to analyze

time series. In this context, the characteristics are the result of data acquisition by a satellite instrument.

The study of star brightness in time (light curves), together with data such as size and distance to its star allows this exoplanet to be characterized.

Classic periodogram analysis requires regularly spaced data, so techniques have been developed to address this limitation. The Lomb-Scargle periodogram (LSP) is a generalization formulated as a modification of Fourier Analysis or a least-squares regression (LSR) that allows to find periods in irregularly spaced data (Feigelson and Babu, 2012). Andrešić et al. (2021) report the use of light curve periodograms to classify variable star types.

The machine learning techniques to classify transit signals include the following methods: (i) random forest (Morton, 2012); (ii) AstroNet, a Deep Learning model (Shallue and Vanderburg, 2018), which validated two new exoplanets; (iii) the machine classifiers used to validate 50 new exoplanets (Armstrong et al., 2020); and (iv) ExoMiner a Deep Learning model (Valizadegan et al., 2022) that validated 301 new exoplanets. Valizadegan et al. (2023) has used the multiplicity boost framework for ExoMiner V1.2 and validates 69 new exoplanets for systems with multiple KOIs from the Kepler catalog.

The objective of this article is to identify possible exoplanets in data generated by astronomical missions employing nanosatellite technology. Our proposal is to use Deep Learning models to search among a large amount of data. Amidst the motivations for this proposal are: (i) Future observations from both traditional satellites (e.g. PLATO (Perryman, 2018) and nanosatellites (e.g. CUTE, CubeSpec (France et al., 2023; Bowman et al., 2022)). (ii) The diversity of the phenomena that produce signals confusing with exoplanetary transits such as eclipsing binaries, stellar variability, and instrumental noise (Deeg and Belmonte, 2018); (iii) A priori lack of knowledge of the transit period of exoplanets, which makes their detection difficult on a large amount of data; and (iv) Supportive work on data from ongoing and future astronomical missions for exoplanets research with nanosatellites (Serjeant et al., 2020).

Today, Deep Learning models are frequently preferred for detection and/or classification problems (Aggarwal, 2018). Deep Learning techniques need a great quantity of labeled data (Sarker, 2021/08/18). Because of this disadvantage, it is highly desirable to have a large quantity of data available to train Deep Learning models. However, collecting large amounts of labeled data for supervised learning is expensive and time-consuming due to the operating conditions in new observatories and space missions. Therefore, available data, the more the better, could be used in the training step to increase the effectiveness of the Deep Learning model in detection and/or classification.

The following are the contributions of this work: (i) We created a validated dataset of 3723 examples of synthetic light curves of exoplanet transit, i.e., 597 (16.04%) Planet Candidates, and 3126 (83.96%) Not Planet Candidates. (ii) We designed the AstroNet model based on Deep Learning that can classify Planet Candidates and Not Planet Candidates using the examples of the validated dataset. (iii) The proposed (AstroNet-46) model has been assessed on the basis of metrics such as accuracy, precision, specificity and recall. (iv) Finally, it has been compared with existing deep neural networks like Convolutional Neural Networks (CNN) by Yeh and Jiang (2020) and Yu et al. (2019) (AstroNet Triage).

To achieve this objective, we conducted the following activities: (i) We compared light curve datasets from catalogs and surveys related to exoplanetary transits; (ii) We investigated techniques for representing transit light curves; (iii) We chose models to represent light curves that serve as input for the detection of exoplanets; (iv) We created synthetic light curve data to train a Deep Learning model; (v) We selected Deep Learning models to detect possible exoplanets on the generated data; and (vi) We assessed the models' performance in detecting exoplanets.

The content of this article is structured into a Conceptual Framework section that provides a description of the concepts and terminology of Deep Learning in the context, highlighting the metrics to evaluate performance of Deep Learning models.

The Proposed Solution section presents the available light curves from the BRITe mission, the pre-processing that is applied and the creation of a light curve dataset with synthetic signals from exoplanets, injected onto BRITe data. The last subsection of the Proposed Solution is the application of Deep Learning models on the generated dataset, which includes the selection of models and training, concluded with the search for possible exoplanets in the pre-processed BRITe data.

The Discussion section presents how the pre-processing and the dataset created under a quality criterion are validated using the AstroNet model and finally the implemented models are validated. The article presents the Conclusions section where the procedures and the most relevant results are compiled, along with future work.

2. Conceptual framework

The folding time series light curve representation technique consists of folding the data into a trial period, and applying statistical methods to folded data such as flux or magnitude, depending on the phase. If the phase is measured in cycles between 0 and 1, the phase ϕ is a function of time t according to Eq. (1), where t_0 is the time where the cycle begins (epoch) and p is a given period, in this way $\phi \in [0, 1]$. By plotting the light curve as a flux in function of phase (instead of a flux as a function of time) a phase-folded light curve or phase diagram is obtained (AAVSO, 2010).

$$\phi = \frac{t - t_0}{p} - \lfloor \frac{t - t_0}{p} \rfloor = (\frac{t - t_0}{p}) \bmod 1 \quad (1)$$

In this method, if the test period corresponds to a periodicity of a pattern present in the curve or is close to it and the epoch is correct, the phase-folded curve with $\phi \in [-1, 1]$ (equivalent to concatenating the previous cycle) can reveal that pattern centered in the middle. In this context, the pattern of interest is the exoplanetary transit.

In the search for exoplanets in the curves of the Kepler mission, Shallue and Vanderburg (2018) use phase-folded curves centered on TCE (Threshold Crossing Event), which correspond to sequences in light curves that resemble the signal of an exoplanet in transit (Christiansen, 2012), which could be a real transit or a false positive. These curves allow two representations to be generated by dividing them into intervals (bins) and forming 1D vectors, according to the following procedure:

1. Define a uniform sequence of intervals (bins) in the time axis with width δ and distance λ between centers of the interval.
2. For each interval, calculate the median flux of the points that fall within the interval. The result is a 1D vector per curve (detection model input).

If $\delta = \lambda$ each data falls exactly in an interval. If $\delta > \lambda$ presents data overlap in the intervals. Shallue and Vanderburg (2018) use two values of λ to generate two views of the curve as input of the detection model. The first, called global view, considers λ as a fraction of the TCE period, which generates a fixed size representation of the entire curve. The second, called local view, considers λ as a fraction of the duration of the TCE, which generates a fixed-size representation of a window around the transit. Both representations are used as input to train a neural network with convolutional layers (CNN).

To generate light curves with transit synthetically, it is necessary to have a model that considers the characteristics of the star, the exoplanet and its respective orbit. A relevant aspect of stars is the so-called limb darkening, a phenomenon where the light at the edge of the stellar disk is less intense and reddish than in the center. The reason for this dimming is the temperature gradient of the stellar photosphere, which makes that light rays that escape from the star near the edge to have

a lower temperature compared to the rays that escape from the center. In Claret (2003) it is confirmed that limb darkening is not a linear phenomenon and compiles laws of its modeling with linear, quadratic, square root and logarithmic behavior.

Mandel and Agol (2002) use the quadratic law to propose an analytical formula for the eclipse of a star by an exoplanetary transit, explaining that the limb darkening effect is significant during an eclipse.

2.1. Deep learning models

A neural network can be used to solve supervised learning problems automatically finding a function to predict the correct output Y or desired from an input X . In this type of learning, the output associated with each example of input is known. Supervised learning is of interest when it comes to the classification of input examples X into categories Y (finite set of classes). The category is known as the example label and in the context of this work, an example of a category used by Yu et al. (2019) are Planet Candidate, Eclipsing Binary, Stellar variability and instrumental noise.

The basic unit of a neural network is the neuron, which is a model that applies a function g (typically nonlinear) to a linear combination of the neuron's input values and its trainable parameters. Eq. (2) presents a model with the data x_1, x_2, \dots, x_d that the neuron receives, such as w_1, w_2, \dots, w_d and a threshold b (called bias). The parameters w_i and b are trainable and $g(\xi)$ is the output value of the neuron corresponding to the result of applying g on the pre-activation ξ .

$$g(\xi) = g\left(\sum_{i=1}^d w_i x_i - b\right) \quad (2)$$

The activation function g gives names to the different types of neurons, the simplest being a linear function.

Nonlinear functions are more useful, i.e., the Sigmoidal neuron (Eq. (3)) applicable in scenarios where the aim is to predict the probability of a binary class.

$$g(\xi) = \frac{1}{1 + e^{-\xi}} \quad (3)$$

Another popular alternative in Deep Learning is the rectifying neuron (ReLU) of Eq. (4) that allows reducing the effective complexity of a network.

$$g(\xi) = \max(0, \xi) \quad (4)$$

2.1.1. Convolutional neural networks

A Feed Forward network allows vectorization of layers. The resulting calculation of a layer (l) (called activation $a^{(l)}$) is represented by a vector of length equal to the number of neurons in that layer. This activation is obtained by applying a function σ to the result of multiplying the activation of the layer $a^{(l-1)}$ (previous layer) by a matrix $W^{(l)}$ and subtracting a vector $b^{(l)}$ (Eq. (5)). Since each element of the layer corresponds to a neuron, the matrix W is a matrix of trainable weights such as W_{ij} is the connection weight between neuron j of layer $(l-1)$ and neuron i of layer (l) , while vector b is the vector activation thresholds. The usual thing is that σ is a non-linear function identical for all neurons in that layer. The input layer does not apply any function or contain trainable parameters since its units correspond to the attributes that describe an example X .

$$a^{(l)} = \sigma_l(W^{(l)} a^{(l-1)} - b^{(l)}) \quad (5)$$

The Convolutional Neural Network (CNN) is popular in contexts where there is high dimensionality and neighboring data are highly correlated, forming thus local patterns. Examples of these contexts are computer vision problems, natural language processing, and time series. These networks follow the following design principle:

- **Local connectivity:** unlike dense connections, this connectivity restricts the neighborhood of each neuron to a subset of neurons adjacent to its input, known as the receptive field. In the 1D case, the field size of a neuron is K positions of the input pattern, being able to move the receptive fields of adjacent neurons according to parameter S (stride), where the i th neuron of a convolutional layer is connected to the positions $[S \cdot i, S \cdot i + K)$ of the input pattern. The number of neurons in the convolutional layer depends on the size of the input and the values of K and S . It is possible to implement padding that tries to maintain the input size in the convolutional layer. To do this, in the 1D case it is filled with zeros on each side of the input. If $S > 1$, the result may not maintain the number of neurons. For the 2D case, the neurons of the convolutional layer are organized in two-dimensional arrays considering two stride values and two receptive field size values (one for each axis).
- **Shared weights:** consist of all receptive fields sharing the same parameters (weights), which allows a group of neurons to work on a certain input pattern. This feature can be extended by implementing multiple groups in the same convolutional layer, where each group (referred filters or channels) share its own trainable parameters. In the 1D case, the number of layer parameters depends on the K value, the number of filters and the number of input filters. The activation of a filter is called feature map.
- **Pooling:** given the potential increase in dimensionality due to the use of convolutional layers, a Pooling layer is introduced, whose objective is to reduce the dimensionality of its input. The reduction is achieved by applying a statistic on receptive fields (defined by size and stride). The usual approach is to calculate the output as the maximum value (Max Pooling) of the associated receptive field, while maintaining the same number of filters from the previous layer constant. This type of layer does not have trainable parameters. In practice, dense layers can be combined with convolutional layers so that the feature map of the last convolutional layer contains the most relevant features for the task and are used as input to a dense Feed-Forward architecture. The union between a convolutional layer and a dense one is done with an intermediate layer called flatten, which takes the input filters and reorganizes them into a one-dimensional output.

2.1.2. Training

In a classification problem, the usual thing is that the number of neurons in the output layer is equal to the number of categories or classes of the problem (or one less), and that the prediction for each neuron is a numerical value associated with the probability that the input example belongs to that particular class, that is, the output of the network is a probability distribution over the classes. This is achieved with a layer whose activation function is softmax. If a valid distribution is sought and the number of classes is 2, it is enough to use a single Sigmoidal output neuron.

The task of training a neural network consists of determining the values of its parameters from examples. Coding the labels of each example is the initial step in this procedure to make them compatible with a prediction. The standard method is to perform one-hot encoding of the labels, that is, transform each label into a vector of length equal to the number of classes, assigning a value of 1 in the j th position if the example belongs to that j th class and a value of 0 in the rest of the positions.

Once the labels are encoded, training requires defining a cost function L (loss) to quantify the error of the network in its prediction. The Categorical Cross-Entropy cost function is the most commonly used in classification problems with mutually exclusive classes. If the number of classes is two, the function of cost is called Binary Cross Entropy.

Once the cost function is established, the network is trained to minimize the function by modifying the trainable parameters in the

negative direction of the gradient of L with respect to each parameter. This movement enables the adjustment of a parameter that is proportional to its influence on the objective function. This is achieved through iterations (called epochs, different from the epochs of the phase-folding concept) over the set of training examples. If N is the number of examples to use, the gradient can be calculated for every training example and the results be averaged. The result of considering the objective as minimizing the expected (approximate) loss value with a finite set of examples from among training is known as minimizing the training error (Aggarwal, 2018).

These steps can be vectorized in order to perform the calculations with a batch of training examples of size b , called mini-epoch. Typically, we look for $b < N$, which allows the parameters to be adjusted immediately using the average gradient of the mini-batch without having to wait for the gradients of all available examples. Once all training examples are used (in their respective mini-epoch), an epoch is completed. If $b \ll N$ the procedure is called Stochastic Backpropagation, which makes an approximation of the true gradient.

Neural networks are models that can present the overfitting problem. That is, they may lose the ability to generalize about examples that were not used for training because the training error can decrease arbitrarily by overfitting the parameters based on the training data, and the network performance can be poor at predicting with data it has not seen. Reducing this prediction error is crucial in the task of detecting exoplanets. To prevent this, the available data is separated into two sets: a training set and a testing set. Training multiple models using only the training set and keep the testing set data independent such that this data they are not used to calculate the parameters of the model. To select models while maintaining the independence of the testing set, the training set is separated into two sets: a new training set and a validation set. The objective of using the three sets is to train models with the training set, choose the best model based on its performance on the validation set and finally evaluate its performance on the testing set.

A problem to face is the choice of a non-representative validation set or that when comparing two or more models, one of them has a better performance over validation, but poor performance on the final data. One solution is the use of K -fold Cross Validation, which consists of dividing the data that is not from the testing set into K blocks of equal size to later train the models using the examples of $K - 1$ blocks as training set, considering as validation set the remaining block. Then, the procedure rotates the blocks iteratively so that each block is validated once. At the end of each iteration, the performance of the model is recorded using a metric. Ultimately, the model with the best average performance is chosen and retrained with all the available data not from the test set.

2.1.3. Metrics to evaluate performance in deep learning

Metrics to evaluate the performance of prediction models allow us to understand how close they are to the desired behavior and provide a framework of comparison with other models. For classification problems, it is common to use Precision, Recall, Specificity, Accuracy and AUC-ROC as metrics, detailed below. These metrics are derived from the confusion matrix, which is a table that visualizes the performance of a predictive model. The values represent a True Positive (TP) when a positive value is well predicted as positive, a False Positive (FP) when a false value is wrong predicted as positive, a False Negative (FN) when a positive value is wrong predicted as negative, and True Negative (TN) when a negative value is well predicted as negative. The details of metrics in the classification task are Shallue and Vanderburg (2018):

- **Precision** (also known as Positive Predictive Value): fraction of correct predictions of the model with respect to a class c . It is the fraction of transit signals classified as planets that are true planets. Its values are in the range between 0 and 1, where values

close to 1 indicate that the number of *FP* is very small compared to *TP*. It is calculated according to Eq. (6).

$$PC_{precision} = \frac{TP}{TP + FP} \quad (6)$$

- Recall (also known as Sensitivity, and as True Positive Rate): fraction of examples of a class *c* of interest that the model effectively recognizes. This is the fraction of true planets that are correctly classified as planets. Values close to 1 indicate that there are few positive cases that were misclassified. It is calculated according to Eq. (7).

$$PC_{recall} = \frac{TP}{TP + FN} \quad (7)$$

- Specificity: measures the incidence of negative cases that were correctly classified. Its values are in the range between 0 and 1. Values close to 1 indicate that there are few negative cases that were misclassified. It is calculated according to Eq. (8).

$$NOT_PC_{recall} = \frac{TN}{TN + FP} \quad (8)$$

- Accuracy: represents the incidence between correct predictions (*TN*+*TP*) and total predictions (*TN*+*TP*+*FP*+*FN*). It may not be a good metric if the dataset is not balanced, i.e., the number of examples of each class is too different. Its values are in the range between 0 and 1, and can also be expressed as a percentage. It is calculated according to Eq. (9).

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \quad (9)$$

- AUC-ROC: Area Under The Curve (AUC) Receiver Operating Characteristics (ROC). Considering different decision thresholds, a recall curve (*PCrecall*) v/s *FP* rate ($1 - NOT_PC_{recall}$) can be constructed, then the area under that curve is the AUC-ROC metric, also called AUC. It is the probability that a randomly selected planet will score higher than a randomly selected *FP*.

3. Proposed solution

3.1. Available data

Our task in this article is to use on the public BRITE data³ to detect potential exoplanets. Public light curves are downloaded from the project Wiki,⁴ which hosts information of interest on the BRITE data. The data is organized in a file tree with two main folders: the public curves and the FITS files with full frame images. The light curves are categorized by sectors (64 sectors as of 07/26/2022), each with a unique identifier. For example, “22-Ori-IV-2016_DR5” corresponds to the 4th observation in the Orion sector within the Data Release 5. A script is programmed to (i) search for all the links that host the files, and (ii) download them automatically.

We use the curves of the following Data Release: DR2, DR3, DR4, DR5 and DR6, where data is available for 609 different stars in 4063 files. Each file is a plain text (ASCII) with a .dat extension. The files contain two sections: the header and the data. The header section has metadata associated with the observation (general information about the nanosatellite, instrument, configuration, etc.), followed by the description of the columns present in the data. The most relevant metadata for this research are: SatellID (satellite short ID), SatLauDa

Table 1

BRITE columns present in the data. The number of columns provided by each file depends on the Data Release, but these are the ones used in this work.

Column	Description	Data release
HJD	Heliocentric Julian Date at start of exposure	2,3,4,5,6
FLUX	Signal extracted from image per second [ADU/s]. FLUXAPT for DR6.	2,3,4,5,6
CCDT	CCD Temperature [°C]	2,3,4,5,6
XCEN	Profile center of gravity with respect to raster origin [pixel]	2,3,4,5,6
YCEN	Profile center of gravity with respect to raster origin [pixel]	2,3,4,5,6
JD	Julian Date at start of exposure as listed in FITS header	2,3,4,5,6
PSFC1	PSF blurring coefficient 1	3,4,5,6
PSFC2	PSF blurring coefficient 2	4,5,6
RTSC	RTS column indicator	3,4,5,6
RTSP	RTS pixel indicator	5,6
APERO	Aperture Offset	5,6
APERF	Aperture Out of Raster indicator: 1 = O K, 0 = out-of-raste	5,6

(satellite orbiting period [min]), SatFilte (filter information), ObsMode (observing mode; it can be ch: chopping horizontally along X axis, or cv: chopping vertically), FieldIDn (observation field ID), ReleaseV (release version), ROIsiz (Raster X size [pixel]), ROlysz (Raster Y size [pixel]), ROlxpos (Raster X position [pixel]) and ROlypos (Raster Y position [pixel]). Table 1 shows the columns used in this work and the DR in which the column is present.

Only data created with chopping observation mode is used. That is, those with value ch (chopping horizontally) or cv (chopping vertically) in the ObsMode field of the header, adding a total of 3147 files that are grouped under the name dataBRITE_chop.

3.2. Pre-processing

The BRITE scientific team recommends pre-processing BRITE data before working on light curves due to three main factors (Pigulski, 2018): (i) The number of hot pixels and other chip defects is high and grows over time. (ii) Pointing is sometimes not ideal, resulting in wobble of stars in rasters, and (iii) An additional Charge Transfer Inefficiency (CTI) defect in some regions of the detectors.

The suggested pre-processing pipeline of Popowicz et al. (2017) does not present a code for a direct application, so we programmed it according to the following:

1. Reformatting raw fluxes and time. It uses Eq. (10) to calculate the magnitude using the FLUX or FLUXAPT column (DR6), subtract the value 2,456,000.0 from the HJD column, adding half exposure time. This operation facilitates the visualization and interpretation of the time axis, now in days. Fig. 1 shows a curve with this step applied.

$$Magnitude = -2.5 \cdot \log_{10}(FLUX) + 14.4 \quad (10)$$

2. Remove extreme outliers for each column ('Magnitude', 'XCEN', 'YCEN', 'CCDT', 'PSFC1', 'PSFC2', 'RTSC', 'RTSP', 'APERO'). In this article we use Sigma Clipping with a $\alpha = 3$ for all columns. Sigma Clipping is an algorithm used in outlier rejection consisting of the following steps:

- (a) Define α .
- (b) Calculate the standard deviation (σ) and median (m) of data.

³ Based on data collected by the BRITE Constellation satellite mission, designed, built, launched, operated and supported by the Austrian Research Promotion Agency (FFG), the University of Vienna, the Technical University of Graz, the Canadian Space Agency (CSA), the University of Toronto Institute for Aerospace Studies (UTIAS), the Foundation for Polish Science Technology (FNiTP MNiSW), and National Science Centre (NCN).

⁴ <http://brite-wiki.astro.uni.wroc.pl/bwiki/doku.php?id=start>

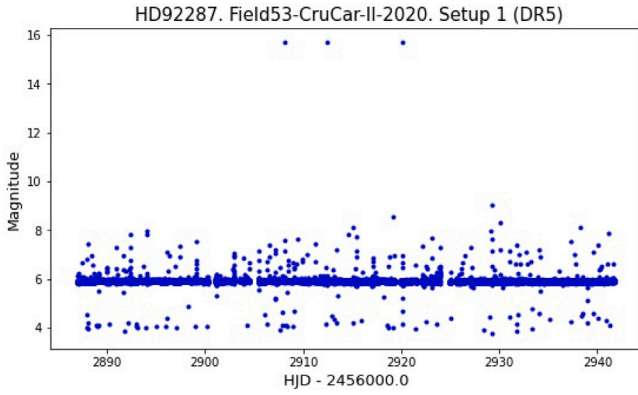


Fig. 1. A light curve of HD92287 resulting from the reformatting of the raw fluxes and the time.

- (c) Remove all points that are smaller or larger than $m \pm \alpha \cdot \sigma$ (outliers).
- (d) Repeat (b) and (c), until reaching exit criteria (typically, a fixed number of iterations).

Due to the nature of the observations in mode chopping, the columns 'XCEN' and 'YCEN' are special cases. In both cases, the values are grouped into two clusters, where the removal of outliers is carried out for each cluster, separately. The separation boundary between both clusters is defined at 0.5 ROIsiz for the horizontal case and 0.5 ROIsiz for the vertical. Both values ROIsiz and ROIsiz are provided by the header and are not necessary for DR6 data since the border of separation is zero.

3. Remove outliers per satellite orbit for each column. The original data does not indicate which data corresponds to what satellite orbit, so we implement the following naïve method to separate the data into orbits:

Naïve method:

- (a) Assume circular satellite orbit.
- (b) Assume that data collection can be carried out in at most half A satellite orbit.

As the BRITE mission has a geocentric orbit, there is no problem applying the naïve method, because it does not work with heliocentric orbit missions. If the satellite orbital period changes over time with respect to the orbital period reported by the team in charge of the satellite, the naïve method is not able to correctly separate the orbits. This difference can be detected by searching the orbital period reported by an external service (e.g., n2yo⁵). This method allows to separate orbits in a comprehensive way without knowing the details of the real orbit or the orbit section used during data collection, by only requiring the satellite orbital period, provided in the header or by Pablo et al. (2016). Figs. 2 and 3 show the diagrams of the proposed method. It can be assured that if data collection begins between A and B, during the next Δ min, the next orbit will not begin. Eq. (11) shows the last orbit index $last_orbit$. Δ is half $orbital_period$.

$$last_orbit = \text{int}\left(\frac{\max(\text{Time}) - \min(\text{Time}) + \Delta}{orbital_period}\right) \quad (11)$$

4. Remove worse orbits, those with greater dispersion measured by standard deviation with respect to any column, using the naïve method of Figs. 2 and 3.

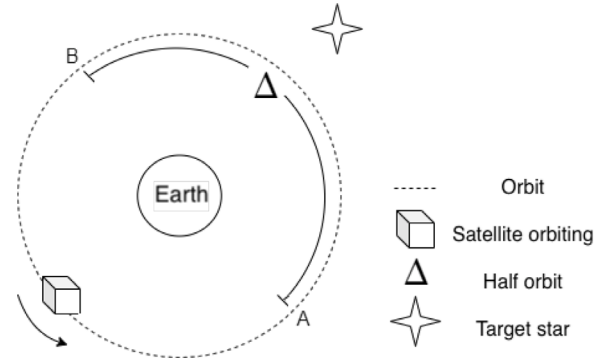


Fig. 2. Orbital diagram of the proposed naïve method.

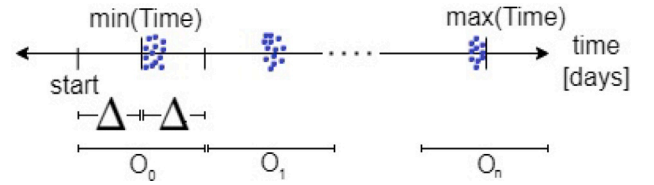


Fig. 3. Data distribution per orbit in the proposed naïve method.

5. Decorrelation: the original magnitudes are correlated with the temperature of the camera, the position of the centroid, orbital phase and other parameters. Correlation is the most significant instrumental effect in BRITE photometry. Popowicz et al. (2017) suggest a 1D and 2D procedure for decorrelation, considering that the first has a greater effect than the latter. We apply the 1D decorrelation considering a parameter (column) P with the following steps:

- (a) consider the Magnitude data as a function of the parameter P .
- (b) fit a curve that describes the dependence with Akima interpolation (library scipy⁶).
- (c) correct the correlation by subtracting the fitted function. It is the programmer's decision whether or not to remove the outliers on the result.

These steps apply for the parameters 'CCDT', 'PSFC1', 'PSFC2', 'RTSC', 'RTSP', 'APER0' and 'Phase' (for DR2, DR3 and DR4 applies to a subset of these parameters). We incorporate the new parameter 'Phase' to analyze instrumental effects related to the satellite orbital phase. Since this is not provided by the original data, the phase of data j in orbit i is calculated according to Eq. (12) such that $Phase_j \in [0, 1]$.

$$Phase_j = \frac{\text{Time}_j - (\text{start} + i * \text{orbital_period})}{\text{orbital_period}} \quad (12)$$

Akima interpolation works as follows: The class is fitted with two corresponding vectors to the points on the x -axis and on the y -axis ($\text{dimension}(x) = \text{dimension}(y) = 1$). The points of both axes are called *anchor_points*. We use 60 pairs of *anchor_points* equi-spaced on the x axis, which correspond to values of the parameter P to be decorrelated and the y points are the arithmetic mean of the Magnitude column that falls between two consecutive x -axis *anchor_points*.

⁵ <https://www.n2yo.com>

⁶ <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.Akima1DInterpolator.html>

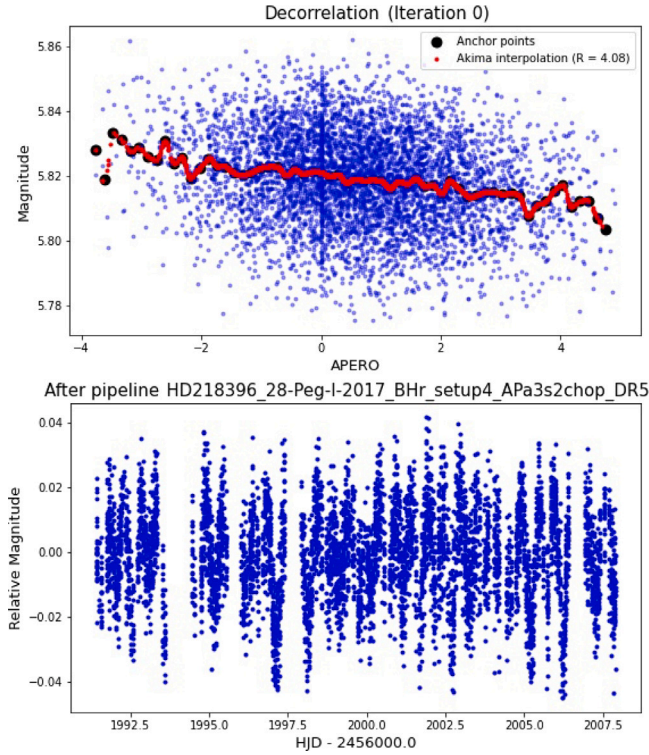


Fig. 4. Top: First iteration of the decorrelation, and Bottom: the final result after 20 iterations.

Once adjusted the Akima function, it is interpolated using the values of the parameter P as input, obtaining a vector defined with the name y_{akima} . At last, we correct the correlation by subtracting y_{akima} from the values of the Magnitude column. This procedure can be applied one or more times for each parameter (iteration). We can obtain different results depending on the order it is applied on the parameters. The criterion for the order of application is to start with the strongest correlation and then continue with the weakest one. To measure the strongest, the parameter R is used and is calculated according to Eq. (13), where V_{pre} and V_{post} are the variances of the Magnitude parameter before and after applying the procedure. The higher the value of R , the stronger is the correlation.

$$R = 100 * (1 - \frac{V_{post}}{V_{pre}}) \quad (13)$$

We suggest finishing the procedure once the R value is less than 0.05 for each parameter. We limit the iterations to a maximum of 20 to limit the execution time. After experimental tests, we discovered that the procedure tends to stagnate in a single parameter. A parameter has always the lowest value of R , so a heuristic is implemented so as not to correct the correlation of the pair P again if the improvement with respect to the previous iteration of the parameter is less than 5%. Eventually, after each correction we remove the Magnitude outliers using sigma clipping with a factor of 4 (less aggressive), and we remove the worst orbits according to the procedure described above. Fig. 4 (top) shows the first iteration applied to the curve HD218396_28-Peg-I-2017_BHr_setup4_APa3s2chop_DR5, and bottom, the final result after 20 iterations.

We applied the described pipeline to the files dataBRITE_chop, obtaining 3139 light curves. The resulting curves are centered at magnitude zero. Of the total, eight (8) files were not used because they trigger an exception in the pipeline.

3.3. Creation of the dataset

3.3.1. Preparation

The BRITE mission is not designed at all to search for exoplanets, so it does not provide data similar to the TCE that the TESS project provides. Consequently, we adapt the Yeh and Jiang (2020) strategy to inject synthetic transits on the BRITE data with the aim of creating a new synthetic dataset of curves with two classes: PC (Planet Candidate) and NOT_PC (Not Planet Candidate). We create three representations for each example: phase fold, local view and global view, used in the work of Yu et al. (2019) for the search of exoplanets with neural networks on the TESS data. The same concatenated BRITE curve can be used to create multiple examples by varying the injection parameters. We work on the 3139 light curves obtained from the pipeline. We concatenate the curves if (i) they correspond to the same target star, and (ii) they correspond to the same satellite. This process results in 876 light curves.

We decided to use the proportion of PC and NOT_PC examples based on Yu et al. (2019). Consequently, the proportion is 16.03% for PC examples and 83.96% for NOT_PC examples.

We decided to inject the transit directly into the pre-processed curves and then create the three representations, with the purpose of preserving the observation windows and their effect on transit, that is, the transit signal is only present at times when the satellite observes the target star.

An important piece of information for creating the dataset is the stellar radius (R_s) measured in Solar radius [R_{sun}] of each target. In the same way, the planetary radius (R_p) is measured in terms of Jupiters radius [R_{jup}]. We did a crossmatch with three catalogs: TIC and Gaia DR2 using the portal of MAST, and CADARS using the service VizieR.⁷ If R_s is in TIC, we use that value, otherwise we search it in GAIA DR2 and finally in CADARS. But if none of them are present, the following procedure is performed:

1. Use SIMBAD⁸ to obtain the values Vmag, Bmag, plx (parallax) and Teff (Effective Temperature), if they are available.
2. Select a Teff from SIMBAD, TIC or GAIA DR2. If not available, use `pyasl.Ramirez2005().colorToTeff` from PyAstronomy,⁹ class used to calculate Teff with Bmag – Vmag.
3. Crossmatch with GAIA DR3 using the MAST portal to obtain available parallax values.
4. Calculate parallax in [mas] ($1[mas] = 1/1000[arcsecond]$) using data from SIMBAD, TIC or GAIA DR3.
5. Use the above data to calculate the R_s of each star.¹⁰

According to the data of exoplanets discovered in NASA Exoplanet Archive with the transit method, the smallest exoplanetary radius is 0.026 [R_{jup}] and the largest is 2.085 [R_{jup}], so an initial constraint is added to the $R_p \in [0.02, 2.1]$ [R_{jup}]. Preliminary tests show poor results in the local and global views, where it is not possible to visually appreciate curves similar to those presented by Yu et al. (2019) and it is difficult to visually differentiate the PC and NOT_PC examples. To analyze the cause of these results we will check the following: (i) The duration of the transit may be too short compared to the temporal resolution of BRITE, or the injected transit falls outside the observation windows; (ii) There might be trends in the concatenated curves; (iii) If there is too much variability, the transit injection cannot be distinguished sufficiently to generate a local or global view with visually noticeable transit; and (iv) The value of $k = R_p/R_s$ may be too small for BRITE precision. It is useful to consider that the value $(R_p/R_s)^2$ approximates the depth of the transit. The greater the depth,

⁷ <http://vizier.cds.unistra.fr/>

⁸ <https://astroquery.readthedocs.io/en/latest/simbad/simbad.html>

⁹ <https://pyastronomy.readthedocs.io/en/latest/index.html>

¹⁰ <https://cas.sdss.org/dr4/en/proj/advanced/hr/radius1.asp>

the more the transit stands out against the noise and variability of the data.

Consequently, the following restrictions must be respected: (i) Considering an orbital period of the satellites close to 100 min and observation windows of approximately 20 min, the duration of the transit is restricted to a minimum of 1 h. Additionally, the duration of a transit cannot exceed the transit period; (ii) Perform a detrend with the Wotan package (Hippke et al., 2019), taking as reference the configuration used by Krishnamurthy et al. (2021) in the search for transits around Alpha Centauri A and B, using data from the ASTERIA mission. This is a biweight filter and a 0.5 day window; (iii) The smallest R_s and largest R_p is calculated such that the depth $(R_p/R_s)^2 > 0.001$ allowing a reasonable variety of R_p . We decided to use the constraint of $R_s < 3.08[R_{sun}]$ and $0.95[R_{jup}] < R_p < 2.1[R_{jup}]$; and (iv) A minimum number of time points during all the transits of the curve. Experimentally, good results were obtained with a minimum of 700 points.

The number of concatenated curves that satisfy the constraint (iii) is 195, with a total of 137 different objects.

The constraint $R_s < 3.08[R_{sun}]$ indicates the presence of several stellar classes. Using the SIMBAD service, we obtain several classes for the dataset targets, mostly A, B and F types. Considering the constraint of $0.95[R_{jup}] < R_p < 2.1[R_{jup}]$ and transits generated with a $p < 5$ [days], the dataset only contains Hot Jupiter. Therefore, it is appropriate to look for the occurrence of Hot Jupiter in the selected classes. Belezny and Kunimoto (2022) present the occurrence of Hot Jupiter in A, F and G dwarfs based on the primary TESS mission. They present graphs with the population of Hot Jupiter in TESS. For class A the range is 10–22 [R_{earth}] (0.89–1.96 [R_{jup}]), and for class F the range is 10–21 [R_{earth}] (0.89–1.87 [R_{jup}]). Therefore, the dataset constraint of 0.95 – 2.1 [R_{jup}] is close to what is expected for classes A and F.

Simpson et al. (2023) examine the photometry of stars with known exoplanets observed by TESS, where they state that “O- and B-type stars ($> 10,000K$) are rare, and thus not well represented in any former or current photometric mission data”.

3.3.2. Injection

For each concatenated curve that meets the restrictions described above, we attempt to generate the PC and NOT_PC examples following the proportion 16.03% and 83.96%, respectively. For each curve, we attempt an injection a maximum of fifty (50) times with different parameters until obtaining a valid transit. Otherwise, we discard the creation of the PC example and then proceed to create NOT_PC examples of that curve.

A curve with transit is modeled with the quadratic model from the PyTransit¹¹ package, which uses the value of $k = R_p/R_s$ as input, two limb darkening coefficients (μ_1, μ_2), time t_0 , period p in days, a_{OS} (semi-major axis divided by stellar radius),¹² inclination i in radians, eccentricity e , argument of the periastron w and an arrangement of exposure times where the model is evaluated. The values used are obtained with a distribution uniform in the ranges of Table 2, except for limb darkening that uses Gaussian distributions. The value t_0 corresponds to the time where the transit is centered.

To determine the temporal indices that correspond to the brightness drop as a result of transit, times are considered within the ranges $[(t_0 \pm p \cdot j) - 0.5 \cdot \text{transit_duration}, (t_0 \pm p \cdot j) + 0.5 \cdot \text{transit_duration}]$, $j \in \{0, 1, 2, 3, \dots\}$. These indices are replaced in the BRITE curve concatenated by the values of flux corresponding to the indices of the curve with transits modeled. The resulting curve is used to create the three representations. The duration of the transit is obtained with the function `pyasl.transitDuration`¹³ of PyAstronomy.

Table 2

Range of values used in transit injection.

Parameter	Values
p	1–5 [days]
a_{OS}	10–30
R_p	0.95–2.1 [R_{jup}]
i	88–90 [$^\circ$]
μ_1	$\mathcal{N}(\mu, \sigma^2)$, $\mu = 0.3, \sigma = 0.05$
μ_2	$\mathcal{N}(\mu, \sigma^2)$, $\mu = 0.1, \sigma = 0.02$
e	0.0
w	0.0
t_0	Random time in curve

Once an injected curve with period p is generated, we follow the procedure described by Yeh and Jiang (2020) to calculate two extra periods in order to create six additional representations with periods two minutes shorter and two minutes longer than p , that is $p \pm 2/1440$ [days]. Then, for each period, we generate the three representations: phase fold, local view and global view. In this way, if an error is not detected during the process, we obtain nine representations corresponding to three PC examples. Fig. 5 shows an example of injected transits in a light curve (up left) and the three representations: Phase fold centered transit (up right), Global View (bottom left), and Local View (bottom right).

We generate NOT_PC examples in a similar way to PCs: we randomly obtain transit parameters that meet the restrictions on a curve but the transit is not injected, then we create the three representations (phase fold, local view and global view) considering a period p , this time without the two additional periods.

The creation of the dataset on the 195 concatenated curves results in 1803 PC examples and 9616 NOT_PC examples. We store the representations in Numpy¹⁴ .npy files and we store all parameters used for transit injection in a .csv file.

The next step is the validation of the generated examples in order to select a subset of the total to train and to evaluate detection models such that these examples meet the following quality criteria: (i) Get a prediction > 0.5 ; or (ii) At least one of the examples generated from the same period p meets the previous criterion, i.e. belongs to the group of examples generated with the period p and $p \pm 2/1440$ [days]. We select 597 PC examples that meet the quality criteria and we randomly choose 3126 NOT_PC examples to obtain the proportion of 16.04% and 83.96%, respectively.

The selected examples are randomly separated into two subsets: training/validation sets and testing set. We use 10% of the total examples for testing (373 examples), so the training/validation set have 3350 examples.

3.4. Application of models

3.4.1. Selection and training

Once created the dataset, the detection can be tested with Deep Learning models, by implementing CNN architectures used by Yeh and Jiang (2020) and Yu et al. (2019) (AstroNet Triage), whose codes use libraries Keras¹⁵ and Tensorflow,¹⁶ respectively.

Fig. 6 shows the histogram of the parameters used in the generation of the dataset (blue for PC examples and orange for NOT_PC examples). The non-uniformity of the parameter planet radius, period and a_{OS} is due to the restrictions on transit duration and minimum number of points during transit. The parameters for NOT_PC are informed, since they are used to generate the views (Global and Local), despite not having injected transits.

¹¹ <https://pytransit.readthedocs.io/en/latest/>

¹² As a reference the value for the Earth-Sun scenario is 215.03.

¹³ <https://pyastronomy.readthedocs.io/en/latest/pyaslDoc/aslDoc/transitDuration.html>

¹⁴ <https://numpy.org/>

¹⁵ <https://keras.io/about/>

¹⁶ <https://www.tensorflow.org/>

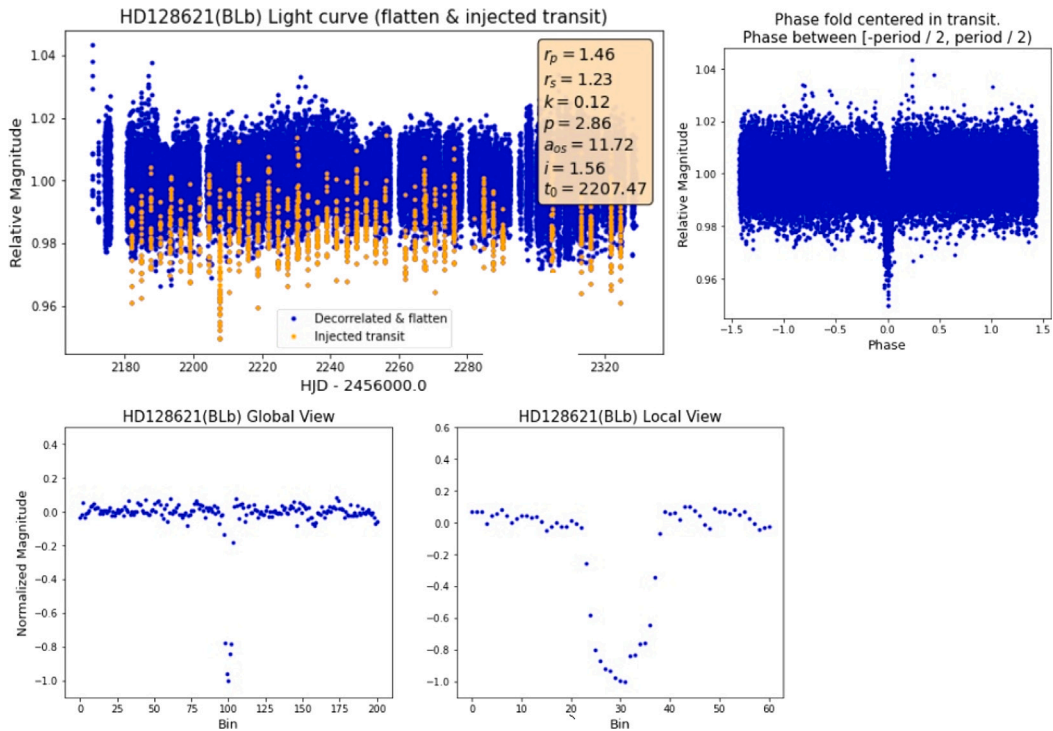


Fig. 5. Up left: Example of generated transits (orange dots). Up right: Phase fold centered in transit. Down: their Global View (left) and Local View (right) representations.

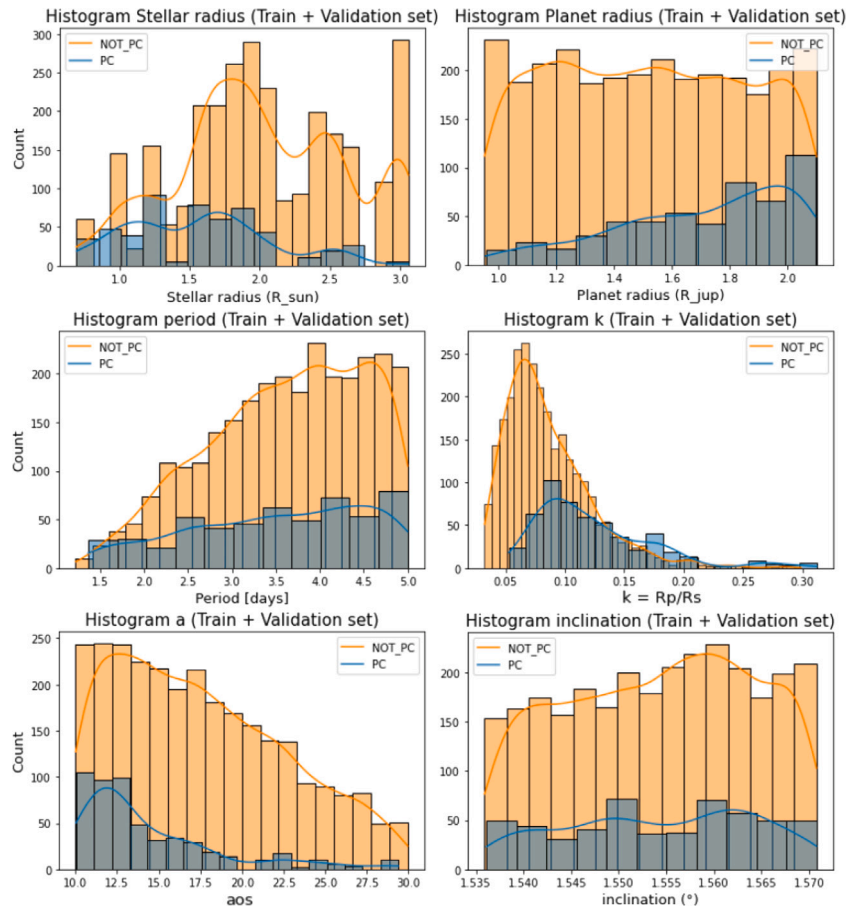


Fig. 6. Histograms with parameters in the training and validation subsets.

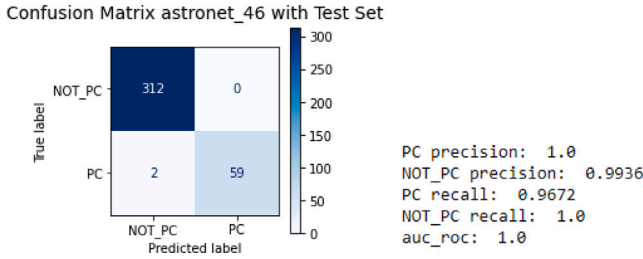


Fig. 7. Confusion matrix and metrics of the AstroNet-46 model on the testing-set.

To simplify this, we implemented the model AstroNet-Triage in Keras (since the original is in Tensorflow) and we modified the input size of the CNN model of Yeh and Jiang (2020) for an input of 201 features (global view). To evaluate the models, a K-Folds Cross Validation is carried out with ten groups to determine the number of epochs to use. For all models, we set the following hyper-parameters and configurations: (i) 3015 training examples and 335 validation examples for each fold; (ii) Loss Binary Cross Entropy; (iii) Adam optimizer with learning rate $1e-05$; (iv) Callback EarlyStopping, with patience 8, monitoring validation loss; (v) Maximum number of epochs: 292; and (vi) Number of examples per gradient update: 70.

The performance of the models is found in Section 4.3. The overall performance is better for the Astronet-Triage model in Keras, so we decided to use both this model and the hyper parameters. The epoch value is set at 46 as it is the average value of the epochs resulting from the ten (10) folds.

The defined model, called AstroNet-46, is trained with the 3350 examples of the training set (3015) and validation set (335), without the step of validation. To measure model performance on data independently of training, the labels of the 373 examples of the testing set are predicted. Fig. 7 shows the confusion matrix and the PC precision, NOT_PC precision, PC recall, NOT_PC recall and AUC_ROC metrics considering a threshold of 0.5, that is, if the model predicts a value ≥ 0.5 it is considered a PC and otherwise NOT_PC. The results close to 1.0 in all metrics suggest a good performance of the model on future data in the task of detecting possible exoplanets. PC precision and NOT_PC recall achieve the optimal result of 1.0, meaning that all predictions labeled as exoplanet are true exoplanets, and all true not exoplanets are well classified as NOT_PC.

3.4.2. Search for potential exoplanets

The next step is to search for possible exoplanets in the pre-processed BRITE data. By focusing on the target stars that have exoplanets already confirmed in NASA Exoplanet Archive (to the date 06/15/2022) and that comply with the radius restriction $R_p < 3.08[R_{sun}]$, we got that the four target stars that meet these conditions are HD 039060, HD 022049, HD 217014 and HD 218396, whose radius are 1.54416, 0.755368, 1.17561 and 1.49338 $[R_{sun}]$, respectively.

We emphasize that the goal of our model is not to find the exoplanets already discovered in these targets, but to look for new signals similar to transits. The only already confirmed exoplanet in the selected targets with a period within the model ranges is 51 Peg b¹⁷ with a period of 4.2307969 days.

The search is carried out by defining the target stars, range of periods to be tested (maximum and minimum), time between periods, and minimum prediction value (score) to record the case as PC. The number of periods to be tested per target is distributed evenly between $[min_period, max_period]$, with a time value of $max_difference$ between consecutive periods of testing. For each period p , different values of epoch are tested, which are uniformly distributed between

Table 3

Target information and predicted parameters for interesting cases.

HD	Name	$[R_{sun}]$	p [days]	$[R_p]$	a_{OS}	t_0
039060	β -Pic	1.54416	1.008688	1.84	13.06	2066.346
022049	eps Eri	0.75536	1.993407	1.84	13.06	1668.834
217014	51 Peg	1.17561	—	—	—	—
218396	HR8799	1.49338	9.982627	1.74	18.49	1941.731
036861	—	1.90	1.995592	1.07	15.36	1371.267

; 4.50

$[min_time, min_time + p]$, with a value around 200 min between consecutive epochs and min_time is the smallest temporal value of the light curve. Furthermore, for each test period, an inclination, planetary radius and semi-major axis divided by stellar radius (a_{OS}) are randomly chosen, within the ranges described in the generation of examples.

We performed two searches for HD 039060, HD 022049, HD 217014 and HD 218396, with a prediction threshold of 0.8. The first search resulted in 75 cases with score > 0.8 in an execution time of 03 : 35 hours with the following configuration: (i) $min_period = 1$ day; (ii) $max_period = 3$ days; (iii) $max_difference = 2.5$ min; and (iv) number of testing periods = 1152.

The second search resulted in 577 cases with score > 0.8 in an execution time of 14 : 32 hours with the following configuration: (i) $min_period = 5$ days; (ii) $max_period = 10$ days; (iii) $max_difference = 5$ min; and (iv) number of testing periods = 1440.

A visual inspection of the cases was conducted to rule out classifications where the signal is clearly not similar to an exoplanet and to identify cases where the signal resembles a candidate. Additionally, other searches were carried out by reducing the prediction threshold to 0.5 for other targets. In this third search we include the target HD036861.

Figs. 8 and 9 present interesting cases whose visual form resembles a possible candidate. Fig. 11 present the result for HD 036861 where the characteristic transit shape is found only in global view. Fig. 12 present the same curve with the parameter a_{OS} manually adjusted to show the transit shape in local view. A detailed inspection of these candidates must be carried out by an astronomer expert in exoplanets.

The first column in Table 3 shows the target HD, the second column is the name of the target, and the third column shows the radius in terms of $[R_{sun}]$. The following four columns show the predicted parameters for interesting cases, the period in days, the R_p , a_{OS} , and t_0 in [HJD-2456000].

During visual inspection, Fig. 10 presents a notable case, where we detect a signal that resemble a variable star, so the model classifies erroneously this variability as an exoplanet. HD 218396 is a candidate to be especially careful of because the model detects a signal similar to a variable star.

4. Discussion

4.1. Previous studies

The study of the BRITE and K2 mission data was explored by Andrešič et al. (2021) with the aim of using artificial neural networks in the classification task of variable stars (in 6 classes: Delta Scuti—young, Detached Eclipsing Binary, Semi-Detached/Contact Eclipsing Binary, Gamma Dor—young, RR Lyrae ab—pulsating and other periodicities). To do this, the authors perform a pre-processing of the data by balancing it into the same number of variable and non-variable stars. The authors analyze the results by classifying with MLP and LSTM with different activation functions, not exceeding 65% accuracy. They conclude that the poor results are probably due to the small-sized intervals of the dataset and the irregular intervals between temporal measurements. It is worth mentioning that up to date (2023) there is more BRITE data available.

¹⁷ <https://exoplanetarchive.ipac.caltech.edu/> (12-26-2023).

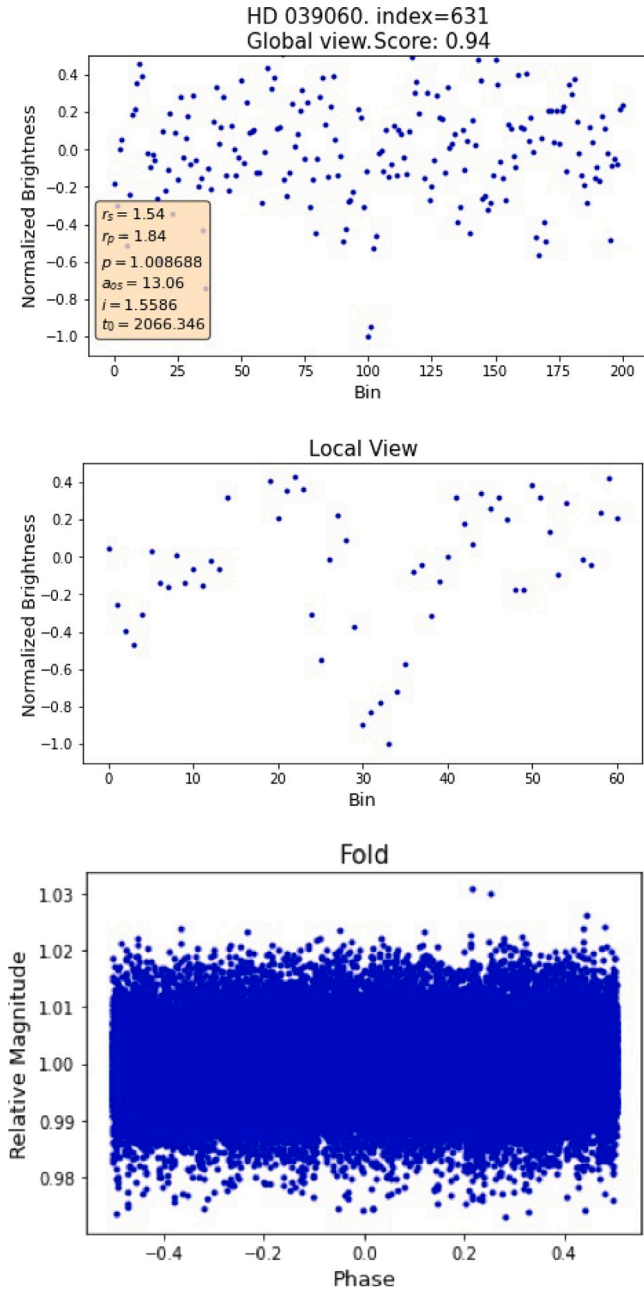


Fig. 8. A case found with exoplanet-like signals in BRITe data, along with predicted parameters. This case corresponds to HD 039060. Up: Global view in normalized brightness. Center: Local View. Bottom: Phase Fold in relative magnitude.

The problem of searching for exoplanets with neural networks on the BRITe mission data is studied by Yeh and Jiang (2020) by injecting synthetic signals on the original data to produce transit curves according to the Mandel-Agol model and detect them with 1D-CNN models. The authors work with DR2 and DR3 of the BRITe mission, considering that observations of the same star made by different nanosatellites correspond to separate light curves, while different curves of a star produced by the same nanosatellite are combined into a single curve. They reduce the dispersion in the curves without making the corrections proposed in Popowicz et al. (2017), resulting in 35 light curves (one per star). Yeh and Jiang (2020) procedure is the following:

1. Divide the curve points into 20 min intervals (bins).
2. Calculate the average flux of each interval.

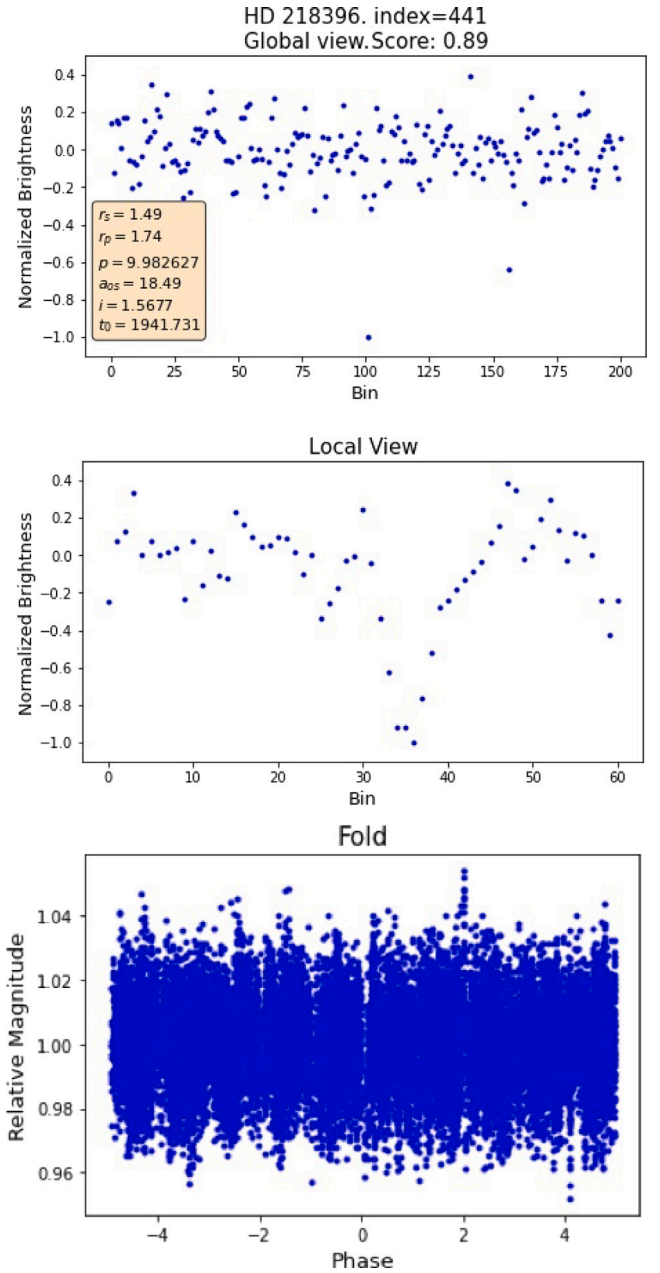


Fig. 9. A case found with signals similar to an exoplanet detected in HD 218396. This signal is found in nearby time epochs.

3. Normalize the previously grouped data with respect to the averages of the observations.
4. Eliminate outliers considering 3 standard deviation.

The parameters included for transit injection are period p , the ratio between the orbital semi-major axis and the radius stellar (a_{0s} between 10 and 30), the ratio between the radius of the planet and the stellar radius (R_{ps} between 0.06 and 0.15), orbital inclination (i between 86° and 90°) and limb darkening coefficients ($\mu_1 = 0.5$ and $\mu_2 = 0.0$). These values are chosen randomly following a uniform distribution in the indicated intervals.

They use the phase-folded method to generate a large number of curves varying the period (trial period) arguing that the majority of possible transit signals present in the original data have been destroyed.

To allow the models to detect transits on curves with a period slightly different from the true period of transit, they created two extra

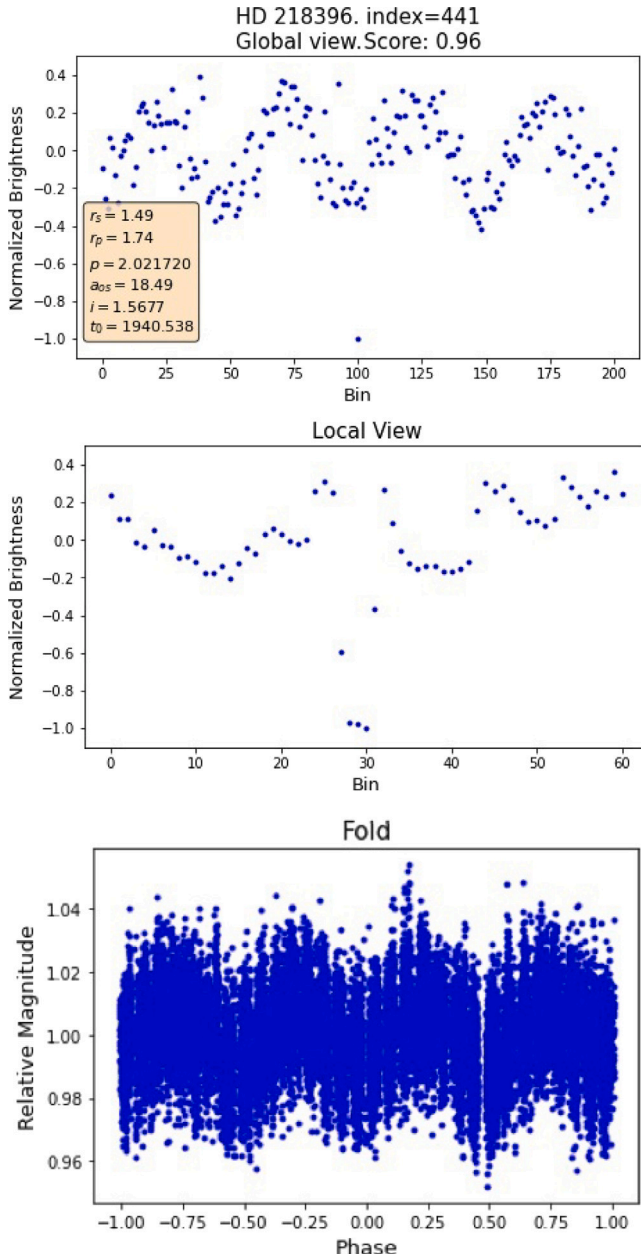


Fig. 10. A case found whose signal resembles a variable star in HD 218396.

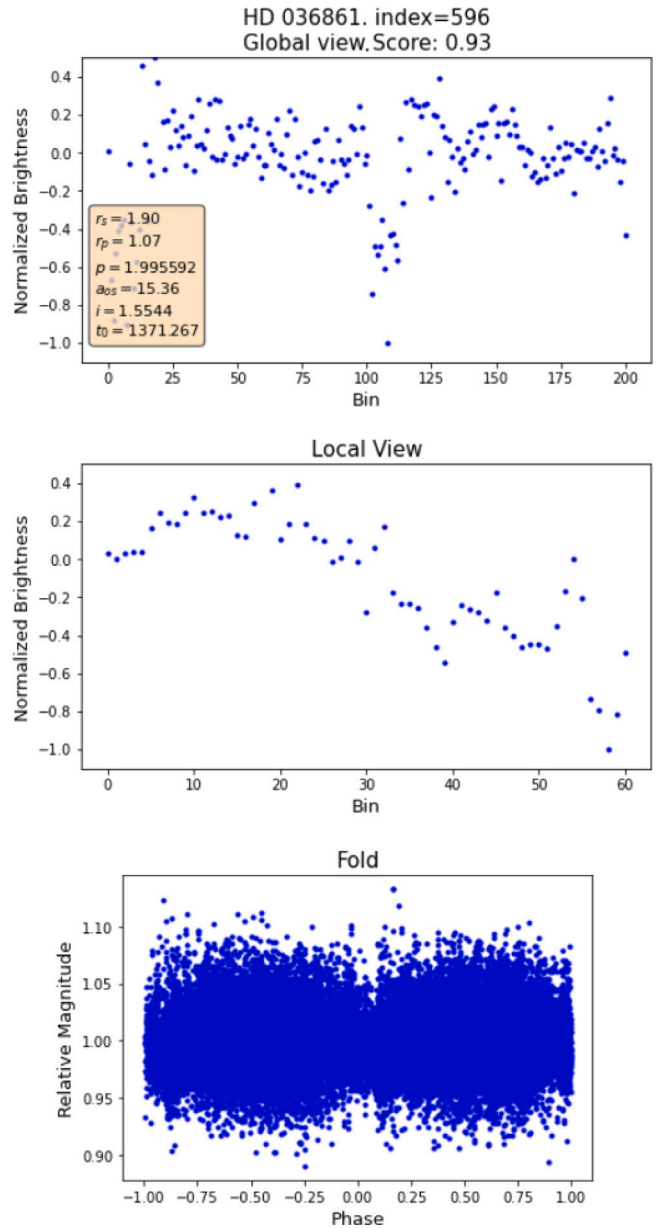


Fig. 11. A case found with signals similar to an exoplanet detected in HD 036861. The characteristic transit shape is not appreciated in local view.

phase-fold curves with periods two minutes shorter and two minutes longer than p , that is, $p \pm 2/1440[\text{days}]$. Half of the data generated correspond to curves with transit and the other half curves without transit, of which 80% is reserved for the training set, 10% for validation set and 10% for testing set.

Regarding the models, they implemented four models with convolutional architecture (1D-CNN). The first of them with the transit detection task considering a period of 1 to 2 days, another model for periods of 2 to 3 days, and so on until the last model for periods between 4 and 5 days. In this way, the period values used in the Mandel and Agol algorithm are uniformly distributed according to the range of each model. Each of these models undergoes five training sessions. The authors used the metrics accuracy, reliability ($PC_{precision}$) and completeness (PC_{recall}) to evaluate the results, obtaining values greater than 99.76%, 99.78% and 99.63%, respectively, for all models. They obtained ten exoplanet candidates, two of them (targets HD37468 and HD186882) with high priority.

The search for exoplanets with neural networks on the TESS mission data is explored by Yu et al. (2019), where they propose the first trained and tested CNN model with real TESS data using the pipeline as a basis and a model proposed by Shallue and Vanderburg (2018) (called AstroNet), in whose original work they identify two new exoplanets in the Kepler data.

Yu et al. (2019) sought to perform two tasks related to the search for exoplanets: (i) create an automatic triage to eliminate all obvious non-planetary signals among TCE, and (ii) create an automatic vetting to differentiate the signals found in (i) in exoplanets or eclipsing binaries.

For task (i) authors consider TCE data labeled in four categories: Planet Candidate (PC), Eclipsing Binary (EB), stellar Variability (V) and Instrumental noise (IS). These labels are assigned under expert judgment following nine classification rules. In total they work with 16,516 TCEs for triage, which includes 493 PCs, 2155 EBs, and 13868 V and IS combined. The data undergoes a random shuffle before being separated into the training set (80%), validation set (10%) and testing set (10%).

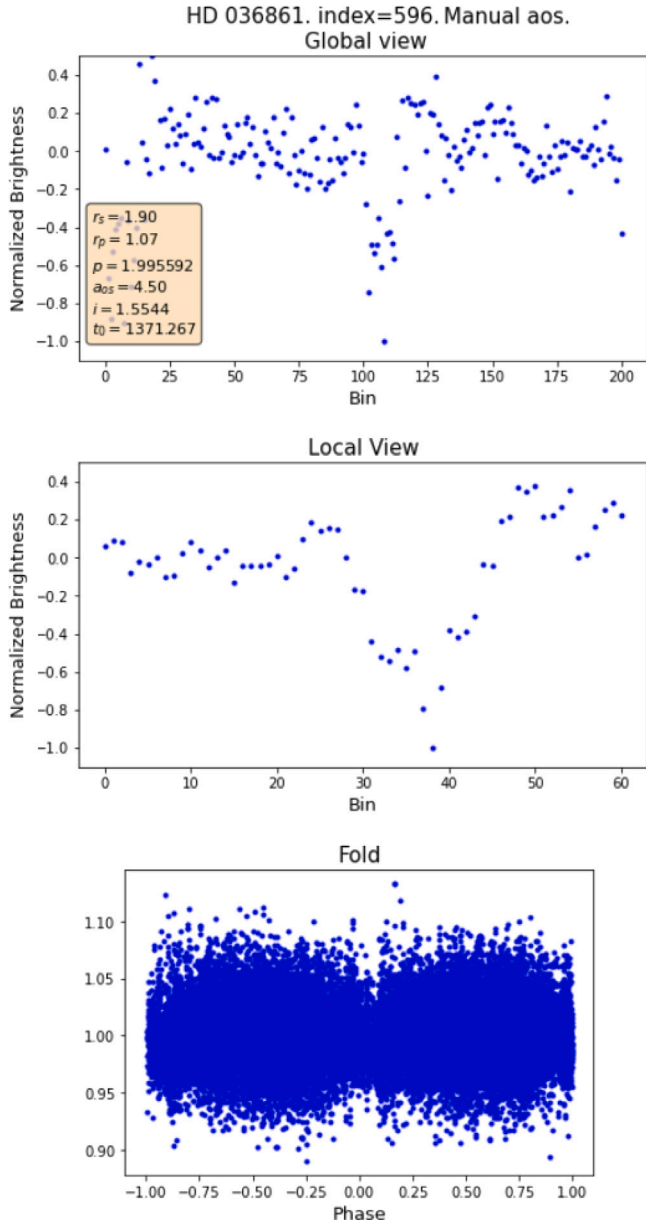


Fig. 12. HD 036861 case with a_{os} parameter manually adjusted to 4.5, where the characteristic transit shape can be seen in local view.

The representation of the curves is carried out by phase folding using the TCE period previously identified, with transit centered on the curves. These curves are then transformed into global view and local view representations.

The authors trained ten identical copies of the AstroNet model (a 1D-CNN implemented with Tensorflow) with different initialization parameters, subsequently averaging its predictions, which is known as model averaging. Using a decision threshold of 0.5, they obtained 97.4% accuracy, 99.2% AUC and 97.0% average precision on the testing set for the triage task.

We validated the pre-processing data recreating the analysis by Sódor and Bognár (2020), which is used to identify significant pulsation frequencies in the star HR 8799 using BRITE data, and comparing them with those detected in MOST data. The authors present a Fourier amplitude spectrum and the frequency components extracted from BRITE photometry. If pre-processing is appropriate, we expect to detect the same frequencies.

We used the public data from HR 8799 for times greater than 1980 [HJD - 2456000] and we obtained the frequencies using a Lomb-Scargle periodogram on the data. Fig. 13 shows the original graph from Sódor and Bognár (2020) at the top and the generated Lomb-Scargle periodogram at the bottom. We obtained the value of $f_1 = 1.97984[1/day]$ from Sódor et al. (2014). Visually, it can be seen that the highest amplitude frequencies coincide perfectly, as does the lowest frequency f_4 and to a lesser extent the frequency $f_1 \cdot \frac{14}{9}$. On the other hand, the frequency f_5 does not match, possibly because the original work uses a greater number of time points. This result validates an appropriate implementation for pre-processing data.

4.2. Validate the dataset

Once the dataset is generated, we verified its quality and we removed the examples that do not meet a quality criterion. To validate the dataset, we re-implemented in Keras the AstroNet-Triage architecture proposed by Yu et al. (2019). Originally, we configured a number of 14000 steps, so considering 13201 examples and a batch size of 64, the number of steps per epoch is $\text{ceil}(\frac{13201[samples/epoch]}{64[samples/step]}) = 207$, therefore we use $\text{ceil}(\frac{14000[steps]}{207[steps/epoch]}) = 68$ as number of epochs. Then we compile with loss binary cross entropy and Adam optimizer with learning_rate $1e-05$. Unlike the original work, we did not apply model averaging. Fig. 14 shows the metric accuracy and the loss resulting from the training and validation. The model AstroNet-46 implemented in Keras receives as input an example of the generated dataset (local and global view representation) and provides a prediction between 0 and 1, indicating that the example belongs to PC or NOT_PC. It can be seen that overfitting occurs from epoch ~ 40 and the accuracy could be improved. However, we decided to continue the work using this trained model since: (i) it corroborates an appropriate implementation in Keras and (ii) this model is used to validate the generated examples of the dataset and not the final classification task.

The creation of the dataset (11419 examples) results in 1803 PC examples and 9616 NOT_PC examples, which are filtered with a criterion quality using predictions from the AstroNet Triage model by Yu et al. (2019).

The model is capable of assigning a value < 0.5 to 100% for the examples labeled as NOT_PC, while only in 360 of the examples labeled as PC we obtained a value > 0.5 , which corresponds to a $\sim 20\%$. Since it is not 100%, the final dataset only included generated PC examples that meet the quality criteria. In this case all the PC examples of the group are considered for the final dataset.

The final validated dataset (3723 examples) consists of the 597 PC examples that meet the criteria of quality and 3126 randomly chosen NOT_PC examples, obtaining a proportion of 16.04% and 83.96%. The dataset is available on Github.¹⁸

4.3. Validation of models

The purpose is to validate the implemented models through the assessment of performance metrics using K -fold Cross Validation. The evaluation helps to select the best model to be used for the detection of exoplanets and to guarantee a good performance for future new data (data that has not been used to train, to test nor to validate the model). For this, we calculate the $PCprecision$, $NOT_PCprecision$, $PCrecall$, $NOT_PCrecall$ and AUC_ROC metrics on the validation set for the two implemented architectures: Yeh and Jiang (2020) and the AstroNet (Yu et al., 2019).

In Figs. 15 and 16 we can appreciate the confusion matrices for the ten folds of the K -fold Cross Validation, along with the average metrics. All metrics in Fig. 15 obtain a value greater than 97% ($PCrecall$), reaching 99% in some metrics ($NOT_PCprecision$, $NOT_PCrecall$,

¹⁸ <https://github.com/alvarofuentesm/synthetic-exoplanet-dataset-1>

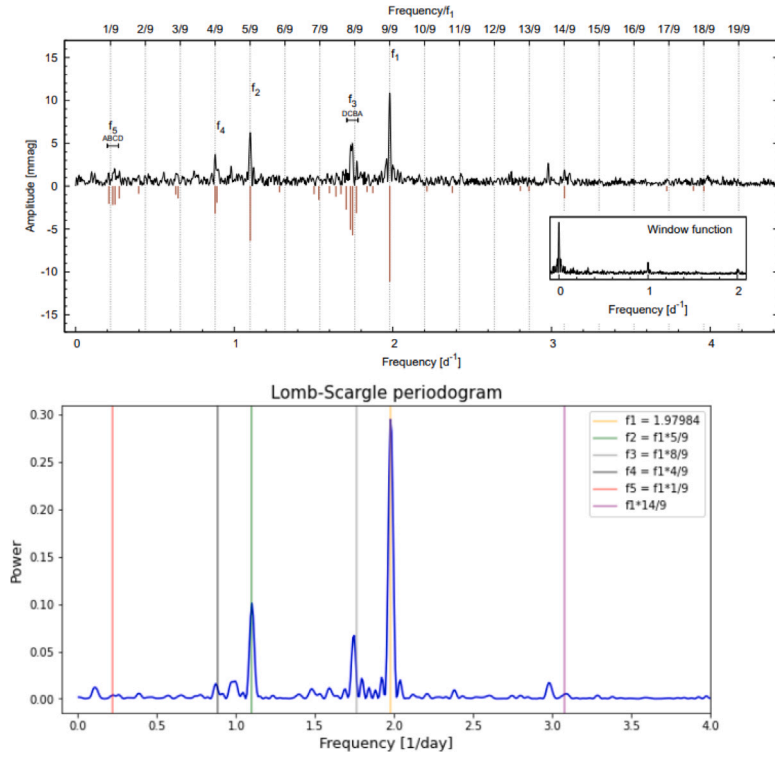


Fig. 13. (top) Calculated Fourier amplitude spectrum of HR 8799 by Sódor and Bognár (2020). (bottom) Lomb-Scargle periodogram generated with the pre-processed data.

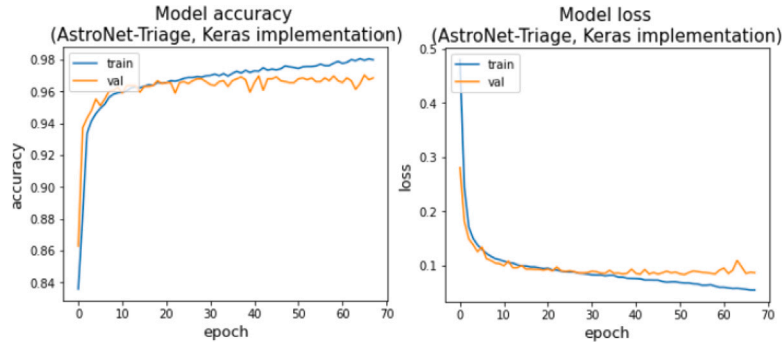


Fig. 14. Training and evaluation metrics for the AstroNet (Yu et al., 2019) model in Keras.

AUC_{ROC}), verifying the good quality of the proposed model. It is observed that the overall values are higher for the AstroNet model in all metrics.

4.4. Comparison with other models

The work in Andrešič et al. (2021) is a classification task of variable stars in six classes, so their work is not comparable with our proposal.

The AstroNet Triage model presented in Yu et al. (2019) was first proposed by Shallue and Vanderburg (2018). We re-implemented this model in Keras under the name AstroNet Keras, trained with the data used by Yu et al. (2019).

Yeh and Jiang (2020) presented four CNN architectures for searching exoplanets in BRITE data, where each architecture is used to train five models (under the name A, B, C, D and E). The four architectures have one 1D input sizes of 144, 216, 288 and 360.

We compared our proposal with AstroNet Keras and Yeh and Jiang (2020) models on the testing-set. Because the input dimensions of

the Yeh and Jiang (2020) architectures differs from our testing set (201 for Global View), the architecture with the smallest size is used, and the input of our dataset is adjusted to the required size (144) by removing the first 28 values and the last 29. The values reported for Yeh and Jiang (2020) correspond to the average of the five models.

Table 4 shows the name of metrics in the first column, the value of each metric obtained by the proposed model (AstroNet-46) in the second column, the Yeh and Jiang (2020) model in the third column and AstroNet Keras in the fourth column.

According to the results shown in Table 4, we can see that the proposal Astronet-46 has higher values in *Accuracy* and *AUC_{ROC}*. Additionally, it obtains the same values as AstroNet Keras in *PCprecision* and *specificity*.

It is important to note that when the dataset is imbalanced in terms of the percentage of PC examples (16.04% in our dataset), the accuracy metric is not particularly informative (Valizadegan et al., 2023).

Some limitations and a plan of action to overcome them in the future are the following: (i) Concerning the creation of the dataset, the first comment is that the restrictions only allow synthetic transits

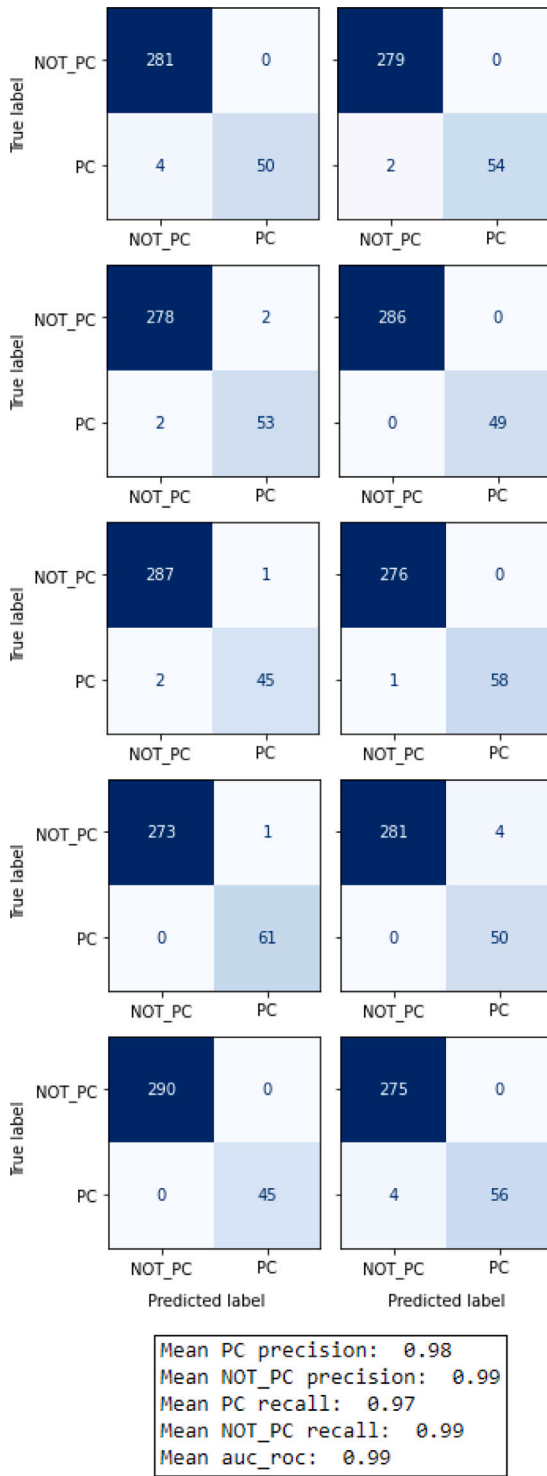


Fig. 15. Confusion matrix and metrics resulting from 10-fold Cross Validation on AstroNet based model.

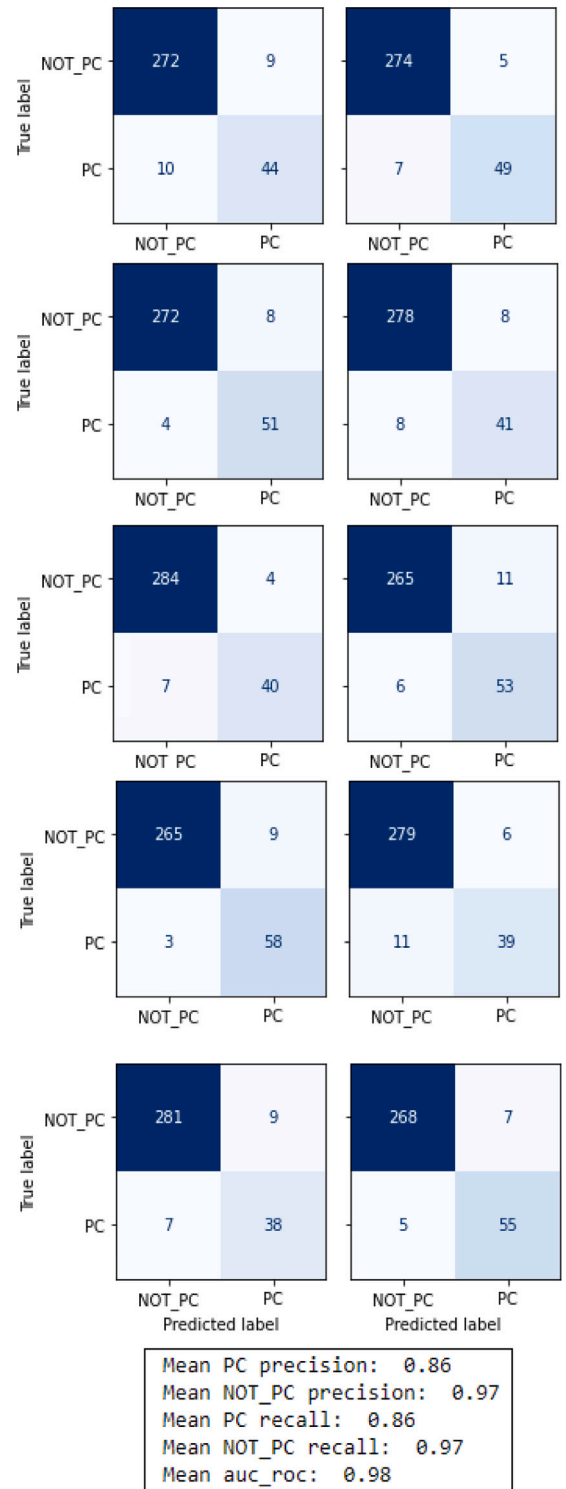


Fig. 16. Confusion matrix and metrics resulting from 10-fold Cross Validation on Yeh and Jiang (2020) architecture based model.

with Hot Jupiter characteristics. A valuable future work is to generate exoplanets with smaller radii and longer periods. On the other hand, the results suggest that at least one class should be added to represent a variable star. (ii) Regarding the search for possible exoplanets, the method requires a test period, which is chosen from an equally spaced list of periods. Although the Deep Learning model allows us to quickly discard periods without a signal in the form of transit, choosing test periods by brute force is still a time-consuming process. The work ahead

is to reduce the period search space. (iii) The method is sensitive to the choice of a_{os} . For example, selecting the a_{os} parameter in Fig. 11 causes the signal to not be seen in the Local View, where a manual modification is necessary to achieve this Local View with the characteristic transit shape (see Fig. 12). (iv) In the second search, we obtained 577 cases that were manually reviewed. It is too large, which motivates us to improve the detection model.

Table 4

Comparison of metrics performance of the proposed model with other models in literature on a testing-set.

Metric	AstroNet-46 proposal	Yeh and Jiang (2020)	AstroNet Keras
Accuracy,%	99.46	26.00	93.56
PCprecision	1.0	0.1816	1.0
PCrecall	0.9672	1.0	0.6065
AUC_ROC	1.0	0.5586	0.9926
NOT_PCrecall (specificity)	1.0	0.1153	1.0

5. Summary and conclusions

The main contribution of this work is the creation of a dataset of light curves with synthetic signals, injected onto the BRITE data. The created dataset is publicly available on Github.¹⁹

Light curves with transit are generated synthetically using the Mandel and Agol quadratic method, which allows modeling a transit with characteristics of the star as its radius and limb darkening coefficients, and the properties of the exoplanet such as its radius and orbital parameters.

The original data is pre-processed as recommended by Pigulski (2018). One contribution is the implementation of a naive method to remove outliers per orbit and remove worse orbits since the original data does not indicate which data corresponds to which orbit. This method can be applied to other space missions.

Additionally, we modified the decorrelation step by limiting the number of iterations to 20 and we used a heuristic to not correct the correlation of a parameter again if the improvement compared to the last iteration of the parameter is less than 5%, adding it to a “tabu list”. We verified that the implementation of pre-processing is appropriate by recreating the Sódor and Bognár (2020) analysis about the star HR 8799, obtaining the same result.

We adapted the Yeh and Jiang (2020) strategy to inject transit signals on the curves. The improvements to their strategy are that it works with a greater number of targets. We applied the pre-processing suggested by the BRITE team, added a step of detrending and restrictions, the main one being the depth of the transit, approximated with $(R_p/R_s)^2 > 0.001$, limiting the number of target stars to those with a radius less than 3.08 Solar radius and exoplanets with a radius between 0.95 and 2.1 Jupiters radius.

A necessary step and a contribution of this work has been to collect and calculate the stellar radius of the targets.

The creation of the dataset resulted in 1803 PC examples and 9616 NOT_PC examples, which were filtered with a quality criterion using predictions from the AstroNet Triage model by Yu et al. (2019). The final validated dataset consists of the 597 PC examples that meet the criteria of quality and 3126 randomly chosen NOT_PC examples, obtaining a proportion of 16.04% and 83.96%, respectively.

With this, we realized a random separation of the selected examples in two subsets: training (80%)/validation (10%) with 3350 examples and testing (10%) with 373 examples.

This article analyzes the detection of potential exoplanets using Deep Learning models on data obtained from the BRITE mission, and with the proposed model (AstroNet-46) we found signals similar to exoplanetary transits in the targets HD 039060, HD 022049, HD 036861 and HD 218396.

To design a Deep Learning model from the created dataset, we compared two CNN architectures: AstroNet Triage (Yu et al., 2019) and the proposal in Yeh and Jiang (2020). We modified the input of the latter to 201 features (Global view). The implemented models have PC precision, Recall and AUC-ROC average values greater than 86%, which

verifies the quality of the models. The overall values are higher for AstroNet model, which obtains average values higher than 97% in all metrics. Consequently, the selected model is the AstroNet architecture with a fixed epoch of 46 (average epoch in the 10-Cross Validation).

To evaluate the proposed model, AstroNet-46 is trained with the training and validation data and the labels of the testing set are predicted with a threshold of 0.5, obtaining 100% precision and 96.72% recall for the PC class, and AUC-ROC of 100%.

5.1. Future work

This work opens the door to the following future works: (i) Increase the number of classes in the synthetic dataset including, for example, eclipsing binaries or variable stars; (ii) To get the limb darkening coefficients of stars, the current method uses a random value within certain fixed ranges, so an estimation could be improved by selecting it through a model or catalog; (iii) Identify a restriction to allow larger radius and then compare the results with the objectives studied by Yeh and Jiang (2020); (iv) Generate synthetic transits for longer periods or smaller radii; (v) Exhaustive search in the rest of the objects, for example, prioritizing those with the least dispersion of magnitudes or those where a PC example was generated²⁰; (vi) Study the transferability of models trained with TESS or Kepler data on the nanosatellite dataset; and (vii) To create a dataset with a modified methodology by first injecting planet signals and then apply pre-processing to the injected data.

CRedit authorship contribution statement

A. Fuentes: Writing – review & editing, Validation, Software, Methodology, Data curation, Conceptualization. **M. Solar:** Writing – original draft, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The link to the dataset is provided into the article.

Acknowledgments

This work was partially funded by the joint project CASSACA-UTFSM (Universidad Técnica Federico Santa María, Chile).

²⁰ HD105211, HD106983, HD110879, HD111123, HD112092, HD114911, HD118716, HD128620, HD128621, HD128898, HD159492, HD160032, HD16970, HD172167, HD173648, HD175362, HD177196, HD178449, HD18331, HD192640, HD195068, HD198639, HD201092, HD201433, HD206267, HD20630, HD211336, HD214168, HD215664, HD217014, HD218396, HD22049, HD23850, HD27459, HD28052, HD32537, HD35296, HD35708, HD36861, HD39060, HD39587, HD4614, HD55892, HD61068, HD6961, HD74375, HD74956.

¹⁹ <https://github.com/alvarofuentesm/synthetic-exoplanet-dataset-1>

References

- AASVO, 2010. Variable star astronomy. Chapter 12: Variable stars and phase diagrams. <https://www.aavso.org/education/vsa>. (Online; Accessed 12 Jun 2022).
- Aggarwal, C., 2018. *Neural Networks and Deep Learning*. Springer.
- Andrešić, D., Šaloun, P., Suchánová, B., 2021. *Intelligent Astrophysics (Emergence, Complexity and Computation, 39)*. Springer.
- Armstrong, D.J., Gamper, J., Damoulas, T., 2020. Exoplanet validation with machine learning: 50 new validated Kepler planets. *Mon. Not. R. Astron. Soc.* 504 (4), 5327–5344. doi:10.1093/mnras/staa2498, arXiv:https://academic.oup.com/mnras/article-pdf/504/4/5327/37975376/staa2498.pdf.
- Beleznyay, M., Kunitomo, M., 2022. Exploring the dependence of hot Jupiter occurrence rates on stellar mass with TESS. *Mon. Not. R. Astron. Soc.* 516 (1), 75–83. doi:10.1093/mnras/stac2179, arXiv:https://academic.oup.com/mnras/article-pdf/516/1/75/45506029/stac2179.pdf.
- Borucki, W.J., 2016. KEPLER mission: Development and overview. *Rep. Progr. Phys.* 79 (3), 036901.
- Bowman, D.M., Vandenbussche, B., Sana, H., Tkachenko, A., Raskin, G., Delabie, T., Vandoren, B., Royer, P., Garcia, S., Van Reeth, T., the CubeSpec Collaboration, 2022. The CubeSpec space mission: Asteroseismology of massive stars from time-series optical spectroscopy. doi:10.48550/arXiv.2208.01533, arXiv e-prints arXiv:2208.01533.
- Centre de Données astronomiques de Strasbourg, 2022. TESS input catalog - v8.0 (TIC-8) : IV/38. <https://cdsarc.cds.unistra.fr/viz-bin/cat/IV/38#/description>. ([Online; Accessed 26 Jan 2024]).
- Christiansen, J.L., 2012. Kepler Q1–Q12 TCE Release Notes. CiteSeer.
- Claret, A., 2003. A new non-linear limb-darkening law for LTE stellar atmosphere models II. *Astron. Astrophys.* 401, 657–660.
- Cryer, J.D., Chan, K.-S., 2008. *Time Series Analysis: with Applications in R*, vol. 2. Springer.
- Dai, Z., Ni, D., Pan, L., Zhu, Y., 2021. Five methods of exoplanet detection. *J. Phys. Conf. Ser.* 2012 (1), 012135. doi:10.1088/1742-6596/2012/1/012135.
- Deeg, H.J., Belmonte, J.A., 2018. *Handbook of exoplanets*.
- Douglas, E.S., Cahoy, K.L., Knapp, M., Morgan, R.E., 2019. Cubesats for astronomy and astrophysics. arXiv preprint arXiv:1907.07634.
- Feigelson, E.D., Babu, G.J., 2012. *Modern Statistical Methods for Astronomy: with R Applications*. Cambridge University Press.
- France, K., Fleming, B., Egan, A., Desert, J.-M., Fossati, L., Koskinen, T.T., Nell, N., Petit, P., Vidotto, A.A., Beasley, M., DeCicco, N., Sreejith, A.G., Suresh, A., Baumert, J., Cauley, P.W., D'Angelo, C.V., Hoadley, K., Kane, R., Kohnert, R., Lambert, J., Ulrich, S., 2023. The colorado ultraviolet transit experiment mission overview. *Astron. J.* 165 (2), 63. doi:10.3847/1538-3881/aca8a2.
- Hippke, M., David, T.J., Mulders, G.D., Heller, R., 2019. Wotan: Comprehensive Time-Series Detrending in Python. *Astron. J.* 158 (4), 143. doi:10.3847/1538-3881/ab3984, arXiv:1906.00966.
- Knapp, M., Seager, S., Demory, B.-O., Krishnamurthy, A., Smith, M.W., Pong, C.M., Bailey, V.P., Donner, A., Di Pasquale, P., Campuzano, B., et al., 2020. Demonstrating high-precision photometry with a CubeSat: ASTERIA observations of 55 Cancri e. *Astron. J.* 160 (1), 23.
- Krishnamurthy, A., Knapp, M., Günther, M.N., Daylan, T., Demory, B.-O., Seager, S., Bailey, V.P., Smith, M.W., Pong, C.M., Hughes, K., et al., 2021. Transit search for exoplanets around alpha centauri A and B with ASTERIA. *Astron. J.* 161 (6), 275.
- Kunitomo, M., Winn, J., Ricker, G.R., Vanderspek, R.K., 2022. Predicting the exoplanet yield of the TESS prime and extended missions through years 1–7. *Astron. J.* 163 (6), 290.
- Lang, K.R., 2013. *Essential Astrophysics (Undergraduate Lecture Notes in Physics)*. Springer.
- Mandel, K., Agol, E., 2002. Analytic light curves for planetary transit searches. *Astrophys. J.* 580 (2), L171.
- Morton, T.D., 2012. An efficient automated validation procedure for exoplanet transit candidates. *Astrophys. J.* 761 (1), 6. doi:10.1088/0004-637X/761/1/6.
- Murphy, S.J., 2012. An examination of some characteristics of Kepler short-and long-cadence data. *Mon. Not. R. Astron. Soc.* 422 (1), 665–671.
- NASA, 2024. NASA exoplanet exploration. <https://exoplanets.nasa.gov/>. (Online; Accessed 26 Jan 2024).
- Pablo, H., Whittaker, G., Popowicz, A., Mochnacki, S., Kuschnig, R., Grant, C., Moffat, A., Rucinski, S., Matthews, J., Schwarzenberg-Czerny, A., et al., 2016. The BRITTE constellation nanosatellite mission: Testing, commissioning, and operations. *Publ. Astron. Soc. Pac.* 128 (970), 125001.
- Perryman, M., 2018. *The Exoplanet Handbook*, second ed. Cambridge University Press.
- Pigulski, A., 2018. BRITTE cookbook 2.0. doi:10.48550/arXiv.1801.08496.
- Popowicz, A., Pigulski, A., Bernacki, K., Kuschnig, R., Pablo, H., Ramaramanantsoa, T., Zocłowska, E., Baade, D., Handler, G., Moffat, A., et al., 2017. BRITTE constellation: Data processing and photometry. *Astron. Astrophys.* 605, A26.
- Ricker, G.R., Vanderspek, R., Winn, J., Seager, S., Berta-Thompson, Z., Levine, A., Villaseñor, J., Latham, D., Charbonneau, D., Holman, M., et al., 2016. The transiting exoplanet survey satellite. In: *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave*, vol. 9904, SPIE, pp. 767–784.
- Sarker, I.H., 2021/08/18. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* 2 (6), 420. doi:10.1007/s42979-021-00815-1.
- Serjeant, S., Elvis, M., Tinetti, G., 2020. The future of astronomy with small satellites. *Nat. Astron.* 4 (11), 1031–1038.
- Shallue, C.J., Vanderburg, A., 2018. Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *Astron. J.* 155 (2), 94.
- Shkolnik, E.L., 2018. On the verge of an astronomy CubeSat revolution. *Nat. Astron.* 2 (5), 374–378.
- Simpson, E.R., Fetherolf, T., Kane, S.R., Pepper, J., Močnik, T., Dalba, P.A., 2023. Variability of known exoplanet host stars observed by TESS. *Astron. J.* 166 (2), 72. doi:10.3847/1538-3881/acda26.
- Smith, M., Donner, A., Knapp, M., Pong, C., Smith, C., Luu, J., Pasquale, P.D., Campuzano, B., 2018. On-orbit results and lessons learned from the ASTERIA space telescope mission.
- Sódor, Á., Bognár, Z., 2020. The planet-host pulsating star HR 8799 as seen by BRITTE. In: *Stars and Their Variability Observed from Space*. pp. 91–92.
- Sódor, Á., Chené, A.-N., De Cat, P., Bognár, Z., Wright, D., Marois, C., Walker, G., Matthews, J., Kallinger, T., Rowe, J., et al., 2014. MOST light-curve analysis of the γ Doradus pulsator HR 8799, showing resonances and amplitude variations. *Astron. Astrophys.* 568, A106.
- Stassun, K.G., Oelkers, R.J., Pepper, J., Paegert, M., De Lee, N., Torres, G., Latham, D.W., Charpinet, S., Dressing, C.D., Huber, D., et al., 2018. The TESS input catalog and candidate target list. *Astron. J.* 156 (3), 102.
- Valizadegan, H., Martinho, M.J.S., Jenkins, J.M., Caldwell, D.A., Twicken, J.D., Bryson, S.T., 2023. Multiplicity boost of transit signal classifiers: Validation of 69 new exoplanets using the multiplicity boost of ExoMiner. *Astron. J.* 166 (1), 28. doi:10.3847/1538-3881/acd344.
- Valizadegan, H., Martinho, M.J.S., Wilkens, L.S., Jenkins, J.M., Smith, J.C., Caldwell, D.A., Twicken, J.D., Gerum, P.C.L., Walia, N., Hausknecht, K., Lubin, N.Y., Bryson, S.T., Oza, N.C., 2022. ExoMiner: A highly accurate and explainable deep learning classifier that validates 301 new exoplanets. *Astrophys. J.* 926 (2), 120. doi:10.3847/1538-4357/ac4399.
- Vanderburg, A., Johnson, J.A., 2014. A technique for extracting highly precise photometry for the two-wheeled Kepler mission. *Publ. Astron. Soc. Pac.* 126 (944), 948.
- Weiss, W.W., Rucinski, S., Moffat, A., Schwarzenberg-Czerny, A., Koudelka, O., Grant, C., Zee, R., Kuschnig, R., Matthews, J., Orleanski, P., et al., 2014. BRITTE constellation: Nanosatellites for precision photometry of bright stars. *Publ. Astron. Soc. Pac.* 126 (940), 573.
- Wright, J.T., Gaudi, B.S., 2012. Exoplanet detection methods. In: *Planets, Stars and Stellar Systems. Volume 3: Solar and Stellar Planetary Systems*.
- Yeh, L.-C., Jiang, G., 2020. Searching for possible exoplanet transits from BRITTE data through a machine learning technique. *Publ. Astron. Soc. Pac.* 133 (1019), 014401.
- Yu, L., Vanderburg, A., Huang, C., Shallue, C.J., Crossfield, I.J., Gaudi, B.S., Daylan, T., Dattilo, A., Armstrong, D.J., Ricker, G.R., et al., 2019. Identifying exoplanets with deep learning. III. Automated triage and vetting of TESS candidates. *Astron. J.* 158 (1), 25.