Automated identification of transiting exoplanet candidates in NASA Transiting Exoplanets Survey Satellite (TESS) data with machine learning methods

Leon Ofman [a,b,*], Amir Averbuch [c,d], Adi Shliselberg [d], Idan Benaun [d], David Segev [d], Aron Rissman [d]

[a] Department of Physics, Catholic University of America, Washington, DC, 20064, USA
[b] NASA GSFC, Code 671, Greenbelt, MD, 20771, USA
[c] School of Computer Science, Tel Aviv University, Tel Aviv, Israel
[d] ThetaRay, 8 Hanagar Street, Hod HaSharon, Israel

ARTICLE INFO

ABSTRACT

A novel artificial intelligence (AI) technique that uses machine learning (ML) methodologies combines several algorithms, which were developed by *ThetaRay, Inc.*, is applied to NASA's Transiting Exoplanets Survey Satellite (TESS) dataset to identify exoplanetary candidates. The AI/ML *ThetaRay* system is trained initially with Kepler exoplanetary data and validated with confirmed exoplanets before its application to TESS data. Existing and new features of the data, based on various observational parameters, are constructed and used in the AI/ML analysis by employing semi-supervised and unsupervised machine learning techniques. By the application of *ThetaRay* system to 10,803 light curves of threshold crossing events (TCEs) produced by the TESS mission, obtained from the Mikulski Archive for Space Telescopes, the algorithm yields about 50 targets for further analysis, and we uncover three new exoplanetary candidates by further manual vetting. This study demonstrates for the first time the successful application of the particular combined multiple AI/ML-based methodologies to a large astrophysical dataset for rapid automated classification of TCEs.

1. Introduction

The Transiting Exoplanet Survey Satellite (TESS) (Ricker et al., 2014) was launched by NASA on April 18, 2018 with the primary objective of all-sky surveying more than 200,000 near-Earth stars in search of transiting exoplanets using high-precision photometry, producing light curves with a 2-minute cadence. The threshold crossing events (TCEs) list for TESS were produced by the Science Processing Operations Center (SPOC) pipeline at NASA Ames from their search of two-minute light curves. The TESS Objects of Interests (TOI) have been released periodically and archived at the Mikulski Archive for Space Telescopes (MAST, https://archive.stsci.edu/). The TOI includes planetary candidates, as well as potential planetary candidates and other astrophysical targets, including false positives, comprising the database used for searching for confirmed exoplanets. As of April 20, 2021 TESS has released 2645 TOIs with 122 confirmed planets, 757 TOIs with planetary radius $r_p < R_{Earth}$ and 745 false positives (see, https://tess.mit.edu/publications/).

Previously, Kepler Space Telescope launched by NASA in 2009 was designed to determine the occurrence frequency of Earth-sized planets. Towards this objective, Kepler observed about 200,000 stars with high photometric precision discovering thousands of transiting exoplanets and exoplanetary candidates (Borucki et al., 2010; Jenkins et al., 2010a; Koch et al., 2010; Christiansen et al., 2012). During the prime missions (2009 May 2 -2013 May 11) Kepler was pointing at a single field of view of about 115 square degrees in the constellations of Cygnus and Lyra. The many periodic signals detected by Kepler were processed using the Kepler Science Processing Pipeline (Jenkins et al., 2010b). They were assembled into a database of TCEs. Direct human input was required to remove false positives and instrumental effects from this database. However, the resulting TCEs database contains data produced by many possible sources, such as eclipsing binaries, background eclipsing binaries and many other possible false alarm sources, in addition to small fraction of exoplanetary candidates (EPCs),

---

and still require considerable analysis for confirmed identification of exoplanets.

Recently, Shallue and Vanderburg (2018) identified transiting exoplanets in Kepler satellite data using Deep Learning (DL) algorithm based on training of convolutional neural networks using the Google-Vizier system (Golovin et al., 2017). Shallue and Vanderburg (2018) trained the neural networks to classify whether a given light curve signal is a signature of a transiting exoplanet with low false positive rate. By using their algorithm, they identify multi-planet resonant chains around Kepler-80. Later, the extended Kepler K2 mission, which starting in Nov. 2013, was designed to use the remaining Kepler capabilities after the completion of the prime mission including the technical failures of the reaction wheels. During this observation phase, the photometric accuracy was reduced, and the pointing varied in different regions of the sky. Nevertheless, Dattilo et al. (2019) used a similar automated technique based on Shallue and Vanderburg (2018) study that is applied to mission data K2 while identifying two previously unknown exoplanets. Multiple ML algorithms were used to validate exoplanets in Kepler data in other recent studies (e.g., Armstrong et al., 2020).

Automated classification methods for transiting exoplanets from TESS data have been developed using machine learning (ML) techniques in several studies (e.g., Zucker and Giryes, 2018; Yu et al., 2019; Osborn et al., 2020) that demonstrate the usefulness and feasibility of this approach with various degrees of improved classification performance. In this paper, we describe an application of novel algorithms, which combine several ML approaches and low rank matrix decomposition, including algorithms that identify anomalies in high dimensional big data by using the augmentation approach. These methods, utilized semi-supervised and unsupervised learning were developed by *ThetaRay, Inc.* (https://thetaray.com/) for uncovering financial crimes, cyber and Internet of Things (IoT) security issues, were applied for transiting EPCs search, reported in this study. By using Kepler data with confirmed exoplanets as part of the algorithm training phase and validation, the *ThetaRay* platform was applied to TESS data yielding three new EPCs out of nearly 11000 TCEs, demonstrating the feasibility and utility of this new platform with unsupervised ML methods.

The paper is organized as follows: Section 2 discusses the ML methods and the details of *ThetaRay* algorithms. Kepler Satellite data ML training are described in Section 3. Section 4 is devoted to the description TESS Satellite Data Analysis and the results of application of *ThetaRay* unsupervised ML algorithm to TESS data. Section 5 is devoted to the discussion and conclusions.

2. Machine learning methods

2.1. The ThetaRay algorithm

In the present study we utilize *ThetaRay* AI-based Fintech algorithms, commercially developed for anomaly detection (financial crimes) in financial institutions, cyber security and IoT for smooth operations of critical infrastructure installations. Since transiting exoplanets light curves are rare and only appear in small number of all observed Kepler or TESS stellar light curves, they are classified as 'anomalies' in our analysis with *ThetaRay* system, utilizing the strengths of its algorithms to identify transiting EPCs in the large number of TCEs. To identify these 'anomalies', or exoplanet light-curves, *ThetaRay*'s algorithms generates a data-driven 'normal' profile of the data ingested, and simultaneously identifies anomalies also called abnormal events, providing forensics that categorizes each event based on its features. This is done autonomously by the algorithm without the need to have rules or signatures. *ThetaRay*'s algorithmic engine utilizes techniques drawn from a wide variety of mathematical disciplines, such as harmonic analysis, diffusion geometry and stochastic processing, low rank matrix decomposition, randomized algorithms in general and randomized linear algebra in particular, geometric measure theory,

manifold learning, neural networks/deep learning, and compact representation by dictionaries. One approach models the data as a diffusion process using Brownian motion of a random walk process to geometrize the data. There is no need for any semantic understanding of the processed data, nor are there any predefined rules, heuristics or weights in the system. The diffused collected dataset is then converted into a Markov matrix through a normalized graph-Laplacian and modeled as a stochastic process that is applied in many dimensions (could reach thousands).

The classification of light curves as exoplanetary candidates in this paper is achieved by using the analytic platform of *ThetaRay* that is described below. This platform processes high dimensional big data to identify anomalous behavior in comparison to a normal profile. This anomaly detection tool is used in the present application for classification of EPCs in TESS TCE database. The normal profile is a training data driven and its generation is explained below. In the present study we used Kepler TCE data as a training dataset as described in Section 3.

Below, we describe the main algorithms that were utilized in the study of identifying anomalies in a big data using augmentation, semi-supervised and unsupervised type algorithms. The same core algorithms for anomaly identification are capable of identifying anomalies in cyber (malware), industrial malfunction (IoT) and financial (crimes) data. The algorithms were applied for the first time to astrophysical data in this study. These algorithms are part of *ThetaRay* (www.thetaray.com) core technology portfolio to fight financial crimes (Shabat et al., 2018a). The algorithms are housed in *ThetaRay* Computational Platform that enables efficient data manipulation and processing, and the reported results were obtained by executing these algorithms on *ThetaRay* platform described below.

2.2. Semi-supervised processing via augmentation: Introduction

For background and context, we describe briefly the *ThetaRay* system current commercial applications that now have been expanded and applied to astrophysical datasets. The *ThetaRay* is designed to provide fast and accurate analytic solutions for identifying emerging risk/crime (classified as anomalies) in financial data, discovering new opportunities, and exposing blind spots within these large, complex high dimensional datasets. These AI-based algorithms radically reducing false positives are uniquely able to uncover "unknown unknowns" (these are threats that one is not aware of, and does not even know that one is not aware of them). *ThetaRay*'s platform covers detection and monitoring of several verticals with current emphasis on financial crimes by supplying an end-to-end solution using an un- and semi-supervised real-time diagnostic, AI based financial crimes detection platform that are based on anomaly detection algorithms. *ThetaRay* provides constructive solutions to anomaly detections challenges via its analytic platform designed for big data, uncovering previously unknown risks, and does so with industry's low false positive rates, in real time enabling fast forensic.

In this project, we assume that some labels of Kepler TCE data, which is a related dataset to TESS TCEs, are given, but, no labels are given for the TESS data. An augmented algorithm, which is considered as a learning method, generates a new data frame based on the provided labels. Then, the new data frame serves as an input to unsupervised algorithms. In this project, we apply 3 unsupervised algorithms to the augmented data: Geometric-based denoted by NY (see Section 2.4.1), algebraic-based denoted by LU (see Section 2.5), a hybrid of LU and NY and Neural network denoted by AE.

The augmentation method is based on Neural Networks. By using a Neural Network-based method, the default network (that can be user-adjusted) consists of one input layer (the analysis data frame), three hidden layers and one output layer. All the layers are connected through "weights" that are automatically tuned during the learning (optimization) process until the network output layer values are close to the values of the provided labels. After optimization, the third

hidden layer becomes the new data frame as well as the input to the unsupervised algorithms that are outlined in Section 2.3 and some of them are described in detail in Section 2.4.

Rule-based technology, which is very popular among anomaly detection tools, is intended for what is known and when one knows what to look for. *ThetaRay*'s detection is achieved by un- and semi-supervised with automatic methods that are not based on rules, patterns, signatures, heuristics, data semantics of the features or any prior domain expertise and provide high detection rate and very low false positives. *ThetaRay*'s methodologies within its Analytics Platform are based on unbiased detection through a series of randomized advanced AI-based algorithms that can process any number of data features and can be explained, justified and anomalies can be traced back to identify features that triggered the anomalies therefore it is not classified as a black box. Thus, the platform enables past tracking of events and features that trigger the occurrence of anomalies. *ThetaRay*'s system operates under the assumption that is not know what to look for or what to ask. This allows their technology to potentially, detect every type of anomaly before the rules are discovered automatically. For efficient processing of the algorithms the system uses off-the-shelf hardware components. Inherent parallelism in the algorithms is implemented with GPU utilization. The platform contains advanced and interactive visualization of the input and output phases of the data analysis. The detection approach is data driven thus; no preexisting models are assumed to exist. This makes this approach universal and generic and thus opens the way for different applications without the introduction of bias, limitations, and unfounded preconceptions into the processing, a property well suited for large astrophysical datasets. Mathematical and physical justification for most of the available algorithms in the system are given below.

The input training data can be enriched by a given limited set of labels. This increases the detection rate and reduces the false alarm rates. This is part of semi-supervised algorithms. Both, semi- and un-supervised algorithms are used in the present analysis. Currently, the platform contains eight different unsupervised algorithms for the data without labels and three different semi-supervised algorithms for the data with partial labels within the detection engine. The results are fused to produce one solution. *ThetaRay*'s system combines the strengths of unsupervised and semi-supervised techniques to identify anomalies in the data. Unsupervised learning assumes that there are no labels to the various data components. Semi-supervised learning frameworks have made significant progress in training machine learning with limited labeled data in image processing domain. Augmented unsupervised learning can be used side-by-side with semi-supervised learning. The augmented algorithms generate a new data frame based on the analysis data frame and the provided labels. The new data frame generated is then the new input for all the unsupervised algorithms selected. Labels are categorized as binaries, with the minority of the labels (known anomalies) marked as "1" and the remainder, which are the majority of unknown cases, assigned "0".

Augmented process enables covering both the known and the unknown with a relative balance between them. The *ThetaRay* system allows for configuration of the underlying input features, algorithms and detection logic at each application. Technically it is a neural network-based process which generates a new data frame based on the input data frame and binary labels provided by the application (in the present case, stellar light-curve data).

2.3. Unsupervised algorithms: General description

NY: This algorithm (see, Fig. 1) is based on diffusion maps (DM) methodology (Coifman and Lafon, 2006a) and it is primarily a non-linear dimension reduction process. The anomaly identification procedure takes place inside the lower dimensional space (manifold) that is determined automatically during the training phase. An out-of-sample extension procedure (Coifman and Lafon, 2006b) is applied to the identification phase for each multidimensional data point, which did not participate in the training phase, to determine whether it belongs to the manifold (low dimensional space — classified as normal) or deviates from it (classified as anomalous).

The NY algorithm, which is based on DM, geometrizes the input training data. DM analyzes the ambient space (training data) and determines automatically where the data actually resides in the embedded space. We can visualize the input training data (ambient space) as a matrix of size $m \times n$ where m is the number of multidimensional data points (number of rows in the matrix) and each row is of dimension n — the number of columns in the matrix. The input data is assumed to be sampled from a low dimensional manifold (embedded space) that captures the dependencies between the observable parameters. DM reduces in a non-linear way the dimension of the ambient space which is the training data. The dimensionality reduction by DM is based on local affinities between multidimensional data points and on non-linear embedding of the ambient space into a lower dimensional space, described as a manifold, by using a low rank matrix decomposition. The non-parametric nature of this analysis uncovers the important underlying factors of the input data and reveals the intrinsic geometry of the data represented by the embedded manifold. This manifold describes geometrically what we classify as the normal profile of the ambient data. Newly arrived multidimensional data points, which did not participate in the training procedure, are embedded into the lower dimensional space by the application of an out-of-sample extension algorithm. If the embedded multidimensional data point falls into the manifold, it is classified as normal otherwise it is classified as abnormal (anomalous). See Section 2.4.1 for more details.

LU: Based on a randomized low-rank matrix decomposition (Shabat et al., 2018b). This algorithm builds a dictionary from the training data. Then, each newly arrived multidimensional data point that is not well described (not spanned well) by the dictionary is classified as an anomalous data point.

The randomized LU (RLU) algorithm is an algebraic approach applied to input matrix A of size $m \times n$ with an intrinsic dimension k smaller than n. k can be computed automatically or given. RLU is a low rank matrix decomposition which enables the identification of anomalies using a dictionary constructed from the training data. RLU forms a low rank matrix approximation of A such that $PAQ \approx LU$ where P and Q are orthogonal permutation matrices, and L and U are the lower and upper triangular matrices, respectively. A dictionary is then constructed according to $D = P^T L$ (T is the transpose of a matrix). Thus, D is a linear combination of the input matrix and a representation of the normal data. It is also used in the identification step to classify newly arrived multidimensional data points that did not participate in the training phase. Thus, a new incoming multidimensional data point x, which satisfies $\|DD^{\dagger}x - x\| < \epsilon$, is classified as normal; otherwise, it is classified as anomalous. Here, $D^{\dagger}$ is the pseudo inverse of D and ϵ is a quantity defined in the training phase. When applied to a matrix A of size $m \times n$, the RLU decomposition reduces the number m of multidimensional data points, resulting in a reduced-measurements matrix of size $k \times n$ where $k < n < m$. Although the algorithm is a randomized, it has been proven in Shabat et al. (2018b) that the probability that the RLU approximation will generate a big error tends to be very small. See Section 2.5 for more details.

DK: The DK Algorithm relies on successive applications of LU and NY. Assume the size of a given training matrix is m data points (rows) by n features (columns). RLU (described in Section 2.5) is
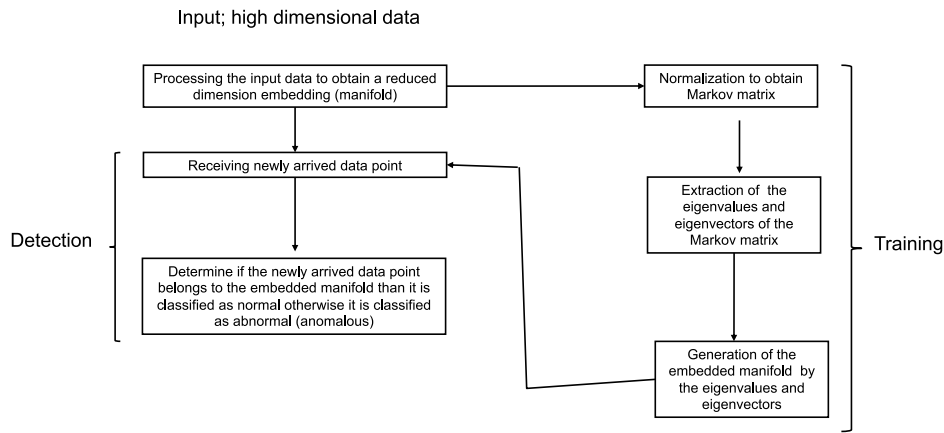
Input; high dimensional data

Fig. 1. NY algorithm: flow chart.

applied to *n*. The size of *n* is reduced substantially through the application of random projection (Johnson and Lindenstrauss, 1984). Then, NY (described in Section 2.4.1) is applied to *n* (dimension) and the matrix is embedded into a lower dimensional space and anomaly identification procedure NY is called in this embedded space.

AE: This is a variational autoencoder (AE) algorithm. AE is machine learning tool designed to generate complex models of data after careful distribution modeling of example data. In neural net language, AE consists of an encoder component and a decoder component. We assume that the input dataset is generated from an underlying unobserved (latent) representation. Given an input dataset, the encoder part of an AE approximates the distribution of the latent variables. Finally, the algorithm sets the distribution parameters of the latent layers in a manner that maximizes the likelihood of generating or reconstructing the input data in the decoder section. As soon as the distribution of the latent variables is approximated, we can sample from this distribution to generate an approximate representation of the input data. Since normality consists of and is defined by most of the data points, those will be well-approximated by the AE, while anomalies will be poorly modeled. Therefore, by comparing the original sample with the reconstructed (generated) data, we can calculate a similarity score that enables us to detect anomalies. The goal is to use the AE as a denoising autoencoder. It allows us to encode our sample into the latent space and then reconstruct it. By comparing the original sample to the reconstruction, we are able to calculate a score that enables us to classify a data point as anomalous data point. Since we plan to use the AE for anomaly detection, we have to calculate the scores for the input and output.

2.4. Unsupervised algorithms: Mathematical description

2.4.1. Diffusion geometry: Background

DM are a kernel-based method for manifold learning that can reveal the intrinsic structures in data and embed them in a low dimensional space. The DM-based approach computes the diffusion geometry. A spectral embedding of the data points provides coordinates that are used to interpolate and approximate the pointwise diffusion map embedding of data.

Manifold learning approaches are often used for modeling and uncovering intrinsic low dimensional structure in high dimensional data. DM is a method that captures data manifolds with random walks that propagate through non-linear pathways in the data. Transition probabilities of a Markovian diffusion process (computational method described below) define an intrinsic diffusion distance metric that

is amenable to a low dimensional embedding. By arranging transition probabilities in a row-stochastic diffusion operator, and taking its leading eigenvalues and eigenvectors, one can derive a small set of coordinates where diffusion distances are approximated as Euclidean distances and intrinsic manifold structures are revealed.

In more details, the NY algorithm uncovers the internal geometry of the input training data denoted as *A*. The use of geometric considerations speeds up significantly the anomaly detection computational time. Next is the theory that supports this approach: the goal is to detect anomalies in *A* and in newly arrived *n*-dimensional data points that did not participate in the training data *A*. During the training procedure, size of *n*, which is also called the dimension of *A*, is automatically reduced. The procedure is called dimensionality reduction. Dimensionality reduction as explained below, is achieved without damaging the quality and the coherency of the data in *A*. More than that, there is no loss of data as explained below. Dimensionality reduction is just a different representation of the training data that automatically without any human intervention reduced the dimension according to the data and uncovers the real dimension where the training data actually resides.

In general, anomaly detection is based on the notion of similarities (or affinities) between the *m* high dimensional data points (these are the rows in the matrix *A*). How one can detect anomalies in this big data efficiently without introducing bias and without damaging the data? For example, when dimensionality reduction of *n* is needed - how to achieve this reduction? The following provides the rationale why geometrization of the training data *A* and tracking the movement of newly arrived data points identify a low dimensional manifold for learning. It is founded mathematically through the preservation of the quality and the integrity (completeness) of the data in *A*.

The assumption is that the processed data is imbalanced: High densities of *n*-dimensional samples (rows in the matrix *A*) represent normal data otherwise the data is classified as anomalous (abnormal) since the majority of the data is normal and thus it is classified as having high density.

Theory: How to find the low dimensional space (manifold)? It is proven that if *A* is sampled from a low intrinsic dimensional manifold, then, as *n* (dimension) tends to infinity, the defined random walk, which travels between all the data samples, converges to a diffusion process over the manifold. This is the key to the processing of *A* as diffusion process that guarantees efficient scan of the data through randomization without introduction of bias. It provides three complementary approaches for dimensionality reduction — diffusion distances between *n*-dimensional samples, randomization and manifold learning — emerge from this observation (theorem): 1. A kernel matrix *B* of size $m \times m$ (huge) is constructed from distances among all the *n*-dimensional samples (rows). The distances are diffusion distances. 2. Random walk is applied to the entries in *B*. This random walk guarantees that there

is no bias between the utilization of the distances in *B*. 3. Diffusion Maps (DM) links between the matrix *B* and a lower dimensional space (manifold) via diffusion processing. The dimension of the embedded manifold represents the reduction of *n*.

Geometrization of the training data — outline description of the approach: The NY algorithm is based on a geometric uncovering of a low dimensional manifold in the ambient space (the original space represented by *A*) by the application of DM to ambient space represented by *A*. The input data is assumed to be sampled from a low intrinsic dimensional manifold that captures the dependencies between the observable parameters (*n*-dimensional features). DM reduces the dimension *n* of the training data. It is based on local affinities between multidimensional data points and on non-linear embedding of the ambient space into a lower dimensional space, described as a manifold, by using a low rank matrix decomposition. The non-parametric nature of this analysis uncovers the important underlying factors of the input data and reveals the intrinsic geometry of the data represented by the embedded manifold. This manifold describes geometrically what we classify as the normal profile in the ambient data. Newly arrived *n*-dimensional data points, which did not participate in the training procedure, are embedded into the lower dimensional space by the application of an out-of-sample extension algorithm. If the embedded *n*-dimensional data point falls into the manifold where most of the normal data reside, it is classified as normal; otherwise, it is classified as abnormal (anomalous). The exchange of data between the ambient space and the manifold, where the detection takes place, does not degrade the coherency and the completeness of the data and preserves the geometrical relations (affinities) between the two spaces — ambient and embedded (manifold).

2.4.2. Diffusion geometry: outline

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a dataset and let $\mathbf{k} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a symmetric point-wise positive kernel that defines a connected, undirected and weighted graph over $\mathcal{X}$. Then, a random walk over $\mathcal{X}$ is defined by the $n \times n$ row-stochastic transition probabilities matrix $P = D^{-1}K$, where K is an $n \times n$ matrix whose entries are $K_{(i,j)} := \mathbf{k}(x_i, x_j)$, $i, j = 1, \dots, n$, and D is the $n \times n$ diagonal degrees matrix whose *i*th element is $\mathbf{d}_{(i)} := \sum_{j=1}^{n} \mathbf{k}(x_i, x_j)$, $i = 1, \dots, n$. The vector $\mathbf{d} \in \mathbb{R}^n$ is referred to as the *degrees vector* of the graph defined by $\mathbf{k}$.

The associated time-homogeneous random walk $X(t)$, is defined via the conditional probabilities on its state–space $\mathcal{X}$: assuming that the process starts at time $t = 0$, then for any time point $t \in \mathbb{N}$ $\mathbb{P}(X(t) = x_j | X(0) = x_i) = P^t_{(i,j)}$, where $P^t_{(i,j)}$ is the (i, j)–th entry of the *t*th power of the matrix *P*. As long as the process is aperiodic, it has a unique stationary distribution $\hat{\mathbf{d}} \in \mathbb{R}^n$ which is the steady state of the process, i.e. $\hat{\mathbf{d}}_{(j)} = \lim_{t \rightarrow \infty} P^t_{(i,j)}$, regardless the initial state $X(0)$. This steady state is the probability distribution resulted from ℓ_1 normalization of the degrees vector $\mathbf{d}$, i.e.,

$$\hat{\mathbf{d}} = \frac{\mathbf{d}}{\|\mathbf{d}\|_1} \in \mathbb{R}^n, \tag{1}$$

where $\|\mathbf{d}\|_1 := \sum_{i=1}^{n} \mathbf{d}_{(i)}$. The *diffusion distances* at time *t* are defined by the metric $\mathbf{D}^{(t)} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$,

$$\mathbf{D}^{(t)}(x_i, x_j) := \left\| P^t_{(i,:)} - P^t_{(j,:)} \right\|_{\ell^2(\hat{\mathbf{d}}^{-1})}$$
$$= \sqrt{\sum_{k=1}^{n} (P^t_{(i,k)} - P^t_{(j,k)})^2 / \hat{\mathbf{d}}_{(k)}}, \quad i, j = 1, \dots, n. \tag{2}$$

By definition, $P^t_{(i,:)}$, the *i*th row of P^t, is the probability distribution over $\mathcal{X}$ after *t* time steps given that the initial state is $X(0) = x_i$. Therefore, the diffusion distance $\mathbf{D}^{(t)}(x_i, x_j)$ from Eq. (2) measures the difference between two propagations along *t* time steps: the first is originated in x_i and the second in x_j. Weighing the metric by the inverse of the steady state results in ascribing high weight for similar probabilities on rare states and vice versa. Thus, a family of diffusion

geometries is defined by Eq. (2), each corresponds to a single time step *t*.

Due to the above interpretation, the diffusion distances are naturally utilized for multiscale clustering since they uncover the connectivity properties of the graph across time. In Bérard et al. (1994) and Coifman and Lafon (2006a) it has been proven that under some conditions, if $\mathcal{X}$ is sampled from a low intrinsic dimensional manifold, then, as *n* tends to infinity, the defined random walk converges to a diffusion process over the manifold.

2.5. Randomized LU decomposition: An algorithm for dictionary construction

A dictionary construction algorithm is presented below. It is based on a low-rank matrix factorization being achieved by the application of the randomized LU decomposition (Shabat et al., 2018b) to a training data. This method is fast, scalable, parallelizable, consumes low memory, outperforms SVD in these categories and works also extremely well on large sparse matrices. In contrast to existing methods, the randomized LU decomposition constructs an under-complete dictionary, which simplifies both the construction and the classification processes of newly arrived multidimensional data points. The dictionary construction is generic and general that fits different applications.

The randomized LU algorithm, which is applied to a given training data matrix $A \in \mathbb{R}^{m \times n}$ of *m* multidimensional data points and *n* features, decomposes *A* into two matrices *L* and *U*. The size of *L* is determined by the decaying spectrum of the singular values of the matrix *A*, and bounded by $\min\{n, m\}$. Both *L* and *U* are linearly independent.

The randomized LU decomposition algorithm (see, Fig. 2) computes the rank *k* LU approximation of a full matrix (Algorithm 1). The main building blocks of the algorithm are random projections and Rank Revealing LU (RRLU) (Pan, 2000) to obtain a stable low-rank approximation for an input matrix *A* that is classified as a training data. In Fig. 2 'II' describes the generation of a dictionaries by calling item I that describes the flow of the randomized LU decomposition. The end of the execution of 'I' means that the training is completed. The dictionaries are the input of 'II' that performs the identification. Newly arrived data point that did not participate in the training is either span (classified as normal) or not spanned by the dictionary (classified as anomalous).

The RRLU algorithm, used in Algorithm 1, reveals the connection between LU decomposition of a matrix and its singular values. Similar algorithms exist for rank revealing QR decompositions (see, for example Gu and Eisenstat, 1996).

Theorem 2.1 (*Pan, 2000*). *Let A be an $m \times n$ matrix ($m \gg n$). Given an integer $1 \leq k < n$, then the following factorization*

$$PAQ = \begin{pmatrix} L_{11} & 0 \\ L_{21} & I_{n-k} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}, \tag{3}$$

holds where L_{11} is a lower triangular with ones on the diagonal, U_{11} is an upper triangular, P and Q are orthogonal permutation matrices. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ be the singular values of A, then:

$$\sigma_k \geq \sigma_{min}(L_{11}U_{11}) \geq \frac{\sigma_k}{k(n-k)+1}, \tag{4}$$

and

$$\sigma_{k+1} \leq \|U_{22}\| \leq (k(n-k)+1)\sigma_{k+1}. \tag{5}$$

Based on Theorem 2.1, we have the following definition:

Definition 2.1 (*RRLU Rank k Approximation denoted $RRLU_k$*). Given a RRLU decomposition (Theorem 2.1) of a matrix *A* with an integer *k* (as in Eq. (3)) such that $PAQ = LU$, then the RRLU rank *k* approximation is defined by taking *k* columns from *L* and *k* rows from *U* such that

$$\text{RRLU}_k(PAQ) = \begin{pmatrix} L_{11} \\ L_{21} \end{pmatrix} (U_{11} U_{12}). \tag{6}$$

where $L_{11}, L_{21}, U_{11}, U_{12}, P$ and *Q* are defined in Theorem 2.1.
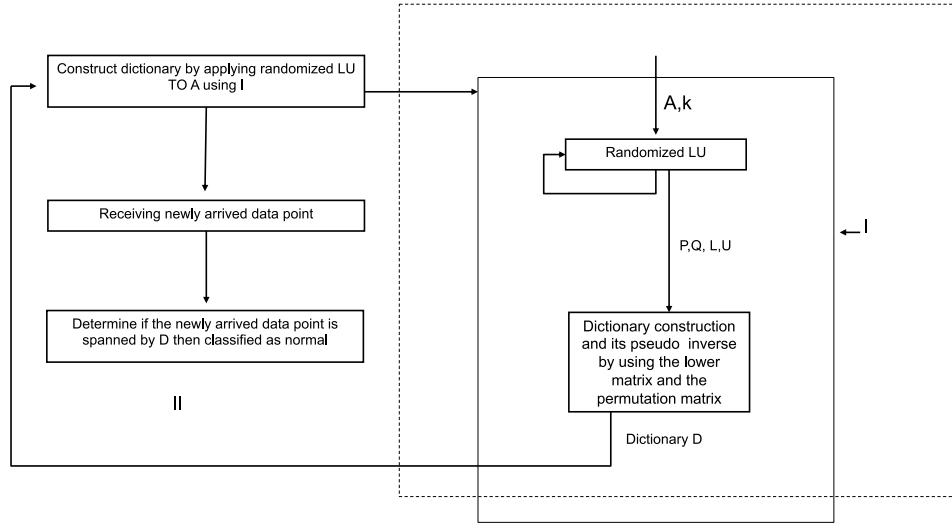
Fig. 2. II calls the construction of a dictionary D via randomized LU decomposition as described in I. The LU algorithm is built from the following steps: The inputs to the algorithm are a matrix A and its rank k (see I). They are submitted to Randomized LU that generates the following outputs: Permutation matrices P and Q and lower and upper triangle matrices L and U, respectively. Then, a newly arrived data point, that did not participate in the training, is either spanned by D therefore classified as normal otherwise it is classified as abnormal (anomalous).

Lemma 2.2 (*Shabat et al., 2018b RRLU Approximation Error*). *The error of the* $RRLU_k$ *approximation of* A *is*

$$\|PAQ - RRLU_k(PAQ)\| \leq (k(n-k)+1)\sigma_{k+1}. \tag{7}$$

Algorithm 1 describes the flow of the RLU decomposition algorithm.

Algorithm 1: Randomized LU Decomposition

Input: Matrix A of size $m \times n$ to decompose; k rank of A; l number of columns to use (for example, $l = k + 5$).

Output: Matrices P, Q, L, U such that $\|PAQ - LU\| \leq \mathcal{O}(\sigma_{k+1}(A))$ where P and Q are orthogonal permutation matrices, L and U are the lower and upper triangular matrices, respectively, and $\sigma_{k+1}(A)$ is the $(k+1)$th singular value of A.

1: Create a matrix G of size $n \times l$ whose entries are i.i.d. Gaussian random variables with zero mean and unit standard deviation.
2: $Y \leftarrow AG$.
3: Apply RRLU decomposition (see Pan (2000)) to Y such that $PYQ_y = L_yU_y$.
4: Truncate L_y and U_y by choosing the first k columns and k rows, respectively: $L_y \leftarrow L_y(:, 1:k)$ and $U_y \leftarrow U_y(1:k, :)$.
5: $B \leftarrow L_y^{\dagger}PA$. ($L_y^{\dagger}$ is the pseudo inverse of L_y).
6: Apply LU decomposition to B with column pivoting $BQ = L_bU_b$.
7: $L \leftarrow L_yL_b$.
8: $U \leftarrow U_b$.

2.6. Randomized LU based classification algorithm

Based on Section 2.5, we apply the randomized LU decomposition (Algorithm 1) to matrix A, yielding $PAQ \approx LU$. The outputs P and Q are orthogonal permutation matrices. Theorem 2.3 shows that $P^T L$ forms (up to a certain accuracy) a basis to A. This is the key property of the classification algorithm.

Theorem 2.3 (*Shabat et al., 2018b*). *Given a matrix* A. *Its randomized LU decomposition is* $PAQ \approx LU$. *Then, the error of representing* A *by* $P^T L$ *satisfies:*

$$\|(P^T L)(P^T L)^{\dagger} A - A\| \leq \left(2\sqrt{2nl\beta^2\gamma^2 + 1} + 2\sqrt{2nl}\beta\gamma \left(k(n-k)+1 \right) \right)$$
$$\sigma_{k+1}(A). \tag{8}$$

Table 1
The effectiveness of the algorithm and its evaluation on Kepler data of size 15,737. False negative in the detection is 2.5%.

True positive	False positive	True negative	False negative	Total
		Training		
2883	52	14	9660	12609
	AUC = 0,957, 80% of the data was used.			
		Detect		
622	126	81	2299	3128
	AUC = 0,921, 20% of the data was used.			

Let x be a multidimensional data point and $D = P^T L$ is a dictionary. The distance between x and the dictionary D is defined by $dist(x, D) \triangleq \|DD^{\dagger}x - x\|$, where $D^{\dagger}$ is the pseudo-inverse of the matrix D. If $dist(x, D) \leq \epsilon$ then x is normal otherwise it is anomalous.

The effectiveness of the algorithm and its evaluation are demonstrated on Kepler data in Table 1 by identifying known planet candidates. The analysis was done in an unsupervised mode using our augmentation methodology. Accuracy is measured by the Area Under ROC Curve (AUC) where ROC is Receiver Operating Characteristic (ROC) curve – a standard evaluation metrics for testing classification model's performance.

2.7. Augmentation for unsupervised learning

The augmented algorithm generates a new data frame when labels are provided. The new data frame then serves as the input to the unsupervised algorithms. The new data frame is defined as a separate learning method. The provided labels are categorized as binaries where the minority of the labels (known as anomalies) are marked as "1" and the remainder, which are the majority of unknown cases, are marked as "0". Two methods are available for achieving augmentation: 1. Based on neural network. 2. Based on geometric component (GC) (Bermanis et al., 2021) analysis combined with SVM. For the neural network-based method, the network consists of one input layer (the data frame), several hidden layers and one output layer. All the layers are connected through "weights" that are automatically tuned during the learning (optimization) process, until the network output layer values are close to the values of the labels provided. After optimization, the third hidden layer becomes the new data frame, as well as the input for the unsupervised algorithms described before. The GCSVM method

first applies the GC algorithm to geometrically down sample the "0" majority class. This is followed by an auto tuned SVM classifier which segregates between the two classes based on the labels provided. The classification information is then used to generate a new data frame for further unsupervised anomaly detection.

Geometric Component (GC) Analysis provides a dictionary-based framework for geometrically driven data analysis, for both linear and non-linear (diffusion geometries), that includes dimensionality reduction, out of sample extension and anomaly detection. The main algorithm greedily picks multidimensional data points that form linear subspaces in the ambient space that contain as much information as possible from the original data. The GC-based diffusion maps appear to be a direct application of a greedy algorithm to the kernel matrix constructed in diffusion maps. The algorithm greedily selects data points from the data according to their distances from the subspace spanned by the previously selected data points. When the distance of all the remaining data points is smaller than a prespecified threshold, the algorithm stops. The extracted geometry of the data is preserved up to a user-defined distortion rate. In addition, a subset of landmark data points, known as dictionary, is identified by the presented algorithm for dimensionality reduction that is geometric-based. It achieves good results for unsupervised learning tasks. The above algorithm is attractive for its simplicity, low computational complexity and tractability.

2.8. Working on ThetaRay's system

We built in *ThetaRay* platform an "analysis chain", which is a multi-staged flowchart, that is composed of three main stages: Data Source, Data Frame and Analysis. The data is organized into data sources and they are uploaded to *ThetaRay*'s platform. We created data frames in the system with wrangling method (where, data wrangling is a process of cleaning, structuring and enriching raw data into a desired format with the intent of making it more appropriate and valuable for analysis) and split the data randomly into train and test components in *ThetaRay* system such that 80% is allocated for training and 20% are allocated for testing. The training procedure generates a data profile and this was fed into different types of analyses using *ThetaRay* Augmented and unsupervised algorithms, to find the best parameters that maximize AUC in each chain. After the analysis and review of these results were completed, the data was processed again after modification and fine tuning of the internal parameters in the system for results improvement. Then, the identification was executed again.

We compared *ThetaRay* algorithms' performance to a leading open-source unsupervised algorithm. They were applied to 17 datasets. Isolation Forest (Liu et al., 2008) was chosen as the leading open-source algorithm. The performance comparison is displayed in Fig. 3 where red represents the performance of Isolated Forest (IF) and blue represents the fused performance (Fu) by *ThetaRay*'s algorithms. The comparison was done between our results that were derived from fusion of the participated algorithms (denoted as Fu) with the Isolated Forest (denoted as IF). The improved performance of the *ThetaRay*'s Fu algorithms over the IF algorithm is evident.

3. Kepler satellite data ML training

We have focused on light curves produced by the Kepler space telescope, which collected the light curves of ~200,000 stars in our Milky Way galaxy for 4 years with continuous 30-min or 1-min sampling. To train the algorithm to identify planets candidates in Kepler light curves, we used a training set of labeled Threshold Crossing Events (TCEs). We obtained all the 15,737 TCEs produced by Kepler and utilized in Google's research deep learning method (Shallue and Vanderburg, 2018; Dattilo et al., 2019), where they used a supervised convolutional neural network machine learning architecture that included 2,202 features: 201 features of "Local View" and 2,001 features of 'global view'. The 'global view' represents the entire light curve and the "Local

View" represents a phase-folded window around the identified transit candidate.

We derived our training set of labeled TCEs from the Autovetter Planet Candidate Catalog for Q1–Q17 DR24 (Catanzarite, 2015; Coughlin et al., 2016) hosted at the NASA Exoplanet Archive (https://exoplanetarchive.ipac.caltech.edu/). We obtained the TCE labels from the catalog's `av_training_set` column, which has three possible values: planet candidate (PC), astrophysical false positive (AFP) and non-transiting phenomenon (NTP). We ignored TCEs with the "unknown" label (UNK). These labels were produced by manual vetting and other diagnostics. We obtained additional data on the TCEs such as planet number, radius of the planet, interval between consecutive planetary transits, etc., from the MAST TESS archive (https://archive.stsci.edu/missions-and-data/transiting-exoplanet-survey-satellite-tess) for data labeling and use in our analysis.

3.1. Features

Feature engineering is the process of using data domain knowledge to create features by manipulating the data through mathematical and statistical relations (for examples, see Section 3.4) of the various components in order to improve the performance of the AI/ML algorithms. The feature engineering process includes deciding which features to develop, creating the features, checking how the features work with the model, improving the features as needed, and going back to deciding on or creating additional data features until the ML/AI algorithm results are optimized. We applied the feature engineering process on our dataset and created new features in addition to the existing features available in MAST in order to provide more information which will quantify various aspects of the data used by the AI/ML algorithm in the present analysis. We produced a total of 424 features that were used for the analysis. We chose the combination of features that provided the best results under the capabilities of *ThetaRay*'s system, validated in the training step. In the feature engineering process, we tested the effectiveness of different combinations of features within the limits of *ThetaRay*'s system.

3.2. Existing features

Additional TCEs Data were downloaded from MAST. We narrowed down the data only to the required fields for the present task, such as the planet number, the radius of the planet, the interval between consecutive planetary transits, etc., and selected the relevant data from all the fields from "Data Columns in the Kepler TCE Table" (https://exoplanetarchive.ipac.caltech.edu/docs/API_tce_columns.html) using the visualization of the variables (especially KDE plots, see below). Below is the description of the variables and labels used in our analysis.

- Unique key — concatenation of Kepler ID and Planet Number. Kepler ID is a target identification number, as listed in the Kepler Input Catalog (KIC). The KIC was derived from a ground-based imaging survey of the Kepler field of view conducted prior to launch. The survey's purpose was to identify stars for the Kepler exoplanet survey by magnitude and color. The full catalog of 13 million sources can be searched at the MAST archive. The subset of 4 million targets found upon the Kepler CCDs can be searched via the Kepler Target Search form.

  – Kepler Input Catalog (KIC) (Brown et al., 2011).
  – MAST archive — http://archive.stsci.edu/kepler/kic10/search.php.
  – Kepler Target Search form — http://archive.stsci.edu/kepler/kepler_fov/search.php.
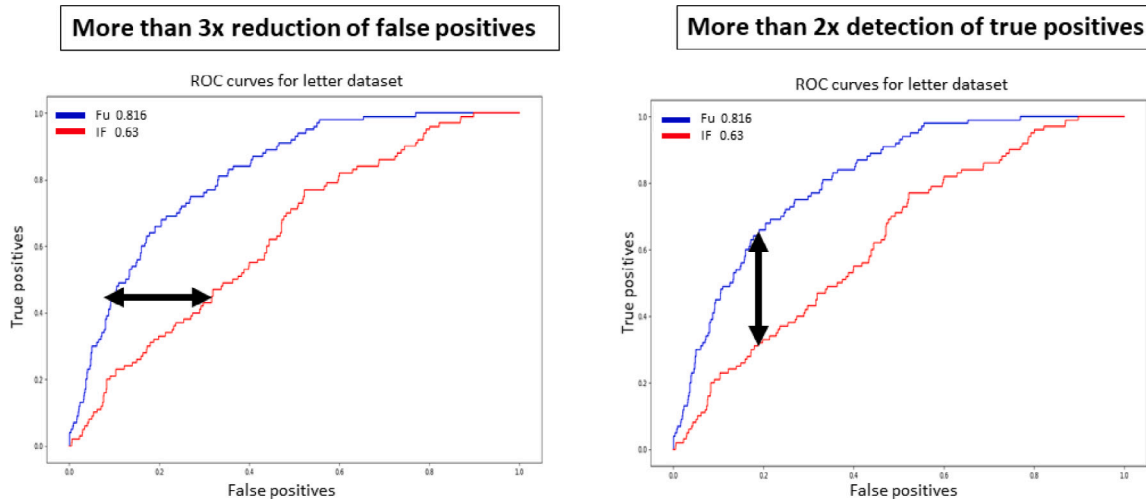
Fig. 3. Comprison of the *ThetaRay*'s and Isolation Forest alogorithms performance. Left panle: More than 3x reduction of false positives. Right panel: More than 2x detection of true positives. On both graphs, the AUC of the fusion (Fu) of *ThetaRay*'s algorithms is 0.816 while Isolation Forest (IF) AUC is 0.63.

- `av_training_set` — Autovetter Training Set Label. If the TCE was included in the training set, the training label encodes what is believed to be the "true" classification, and takes a value of either PC, AFP or NTP. The TCEs in the UNKNOWN class sample are marked UNK. Training labels are given a value of NULL for TCEs not included in the training set. For more detail about how the training set is constructed, see Autovetter Planet Candidate Catalog for Q1–Q17 Data Release 24 (KSCI-19091): https://exoplanetarchive.ipac.caltech.edu/docs/KSCI-19091-001.pdf.
- `tce_prad` — Planetary Radius (Earth radii). The radius of the planet obtained from the product of the planet to stellar radius ratio and the stellar radius.
- `tce_max_mult_ev` — Multiple Event Statistic (MES). The maximum calculated value of the MES. TCEs that meet the maximum MES threshold criterion and other criteria listed in the TCE release notes are delivered to the Data Validation (DV) module of the data analysis pipeline for transit characterization and the calculation of statistics required for disposition. A TCE exceeding the maximum MES threshold is removed from the time-series data and the SES and MES statistics recalculated. If a second TCE exceeds the maximum MES threshold then it is also propagated through the DV module and the cycle is iterated until no more events exceed the criteria. Candidate multi-planet systems are thus found this way. Users of the TCE table can exploit the maximum MES statistic to help filter and sort samples of TCEs for the purposes of discerning the event quality, determining the likelihood of planet candidacy, or assessing the risks of observational follow-up. DV module — http://archive.stsci.edu/kepler/manuals/KSCI-19081-001_Data_Processing_Handbook.pdf
- `tce_period` — Orbital Period (days). The interval between consecutive planetary transits.
- `tce_time0bk` – Transit Epoch (BJD) – 2,454,833.0. The time corresponding to the center of the first detected transit in Barycentric Julian Day (BJD) minus a constant offset of 2,454,833.0 days. The offset corresponds to 12:00 on Jan 1, 2009 UTC.
- `tce_duration` — Transit Duration (hrs). The duration of the observed transits. Duration is measured from first contact between the planet and star until last contact. Contact times are typically computed from a best-fit model produced by a Mandel and Agol (2002) model fit to a multi-quarter Kepler light curve, assuming a linear orbital ephemeris.
- `tce_model_snr` — Transit Signal-to-Noise (SNR). Transit depth normalized by the mean uncertainty in the flux during the transits.

- `av_pred_class` — Autovetter Predicted Classification. Predicted classifications, which are the 'optimum MAP classifications.' Values are either PC, AFP, or NTP.
- `tce_depth` — Transit Depth (ppm). The fraction of stellar flux lost at the minimum of the planetary transit. Transit depths are typically computed from a best-fit model produced by the Mandel and Agol (2002) model fit to a multi-quarter Kepler light curve, assuming a linear orbital ephemeris.
- `tce_impact` — Impact Parameter. The sky-projected distance between the center of the stellar disc and the center of the planet disc at conjunction, normalized by the stellar radius.
- `local_view` — vector of length 201: a "Local view" (phase-folded) of the TCE. It shows the shape of the transit in detail (close-up of the transit event).

3.3. Visualization of Kepler data

We investigated the Kepler data and visualized the variables with Pandas package in Python. For example, we visualize the distributions of the numerical variables per class using KDE (Kernel Density Estimation) plots. In Fig. 4 we show several interesting examples with a gap between the curves labeled 'Planets' and 'Not planets' as identified by *ThetaRay* system and validated by the Kepler data training set. It can be concluded that these features are significant in candidate exoplanet identification and therefore we have included them in the model. If both curves coincide, it can be concluded that the behavior is the same for label 'planets' and 'not planets', and so we chose not to include these features in the model.

Another example of our analysis is demonstrated in the 'heat map', which is basically a color-coded matrix, where a correlation value between the variable of features is used to color each cell of the matrix to represent the relative value of that cell. If there is a high correlation between any variables, the dimension of the data can be reduced. The various features are labeled on the axes. Obviously, the features on the main diagonal that indicate identity correlation are light colored. It is evident from the 'heat map' shown in Fig. 5 that most off-diagonal features are weakly correlated. The only significant off-diagonal correlations are between `av_training_set` — the training labels, i.e., if the TCE was included in the training set, the training label encodes what is believed to be the "true" classification, and `av_pred_class` — predicted classifications, which are the optimum MAP (maximum a posteriori) classifications. In fact, this field does not provide analysis information for the data but is used as forensic feature. The forensic features are not included directly in the analysis
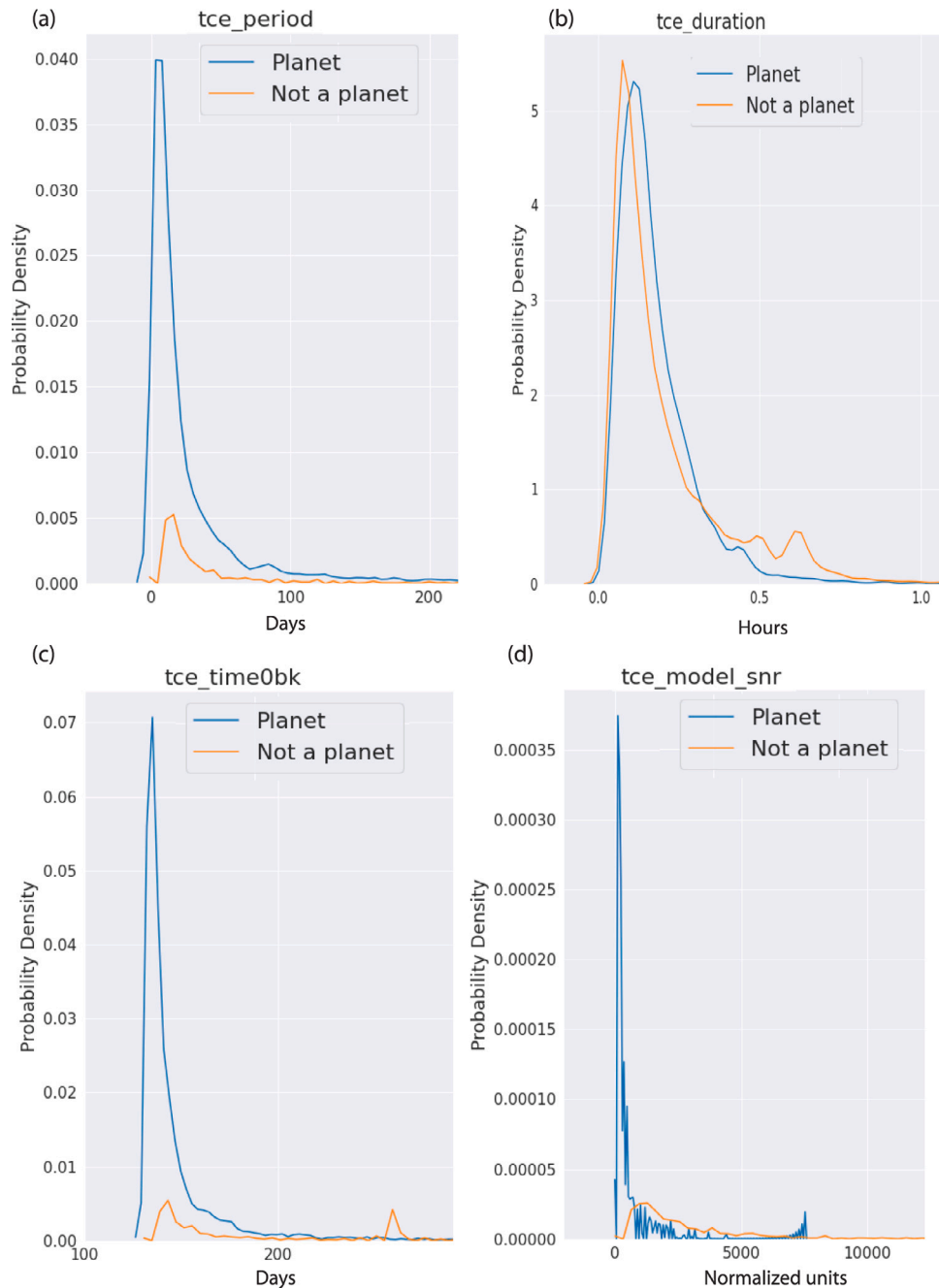
Fig. 4. The distributions of the numerical variables using KDE (Kernel Density Estimation) plots where the blue curves are labeled 'Planet' and the orange curves are labeled ' Not a planet' from Kepler data. When there is significant difference between the curves, it can be concluded that these features are more significant for planet identification and therefore we have included them in the model. If both curves coincide, it can be concluded that the behavior is not statistically different between the two populations. The plotted variables are (a) `tce_period`, (b) `tce_duration`, (c) `tce_time0bk`, (d) `tce_model_snr` (see text for their definitions).

but provide supplementary information about the data useful for the investigation of the analysis. Some artificial correlation is also evident between the `tce_time0bk` — transit epoch (BJD), and `tce_period` — Orbital Period (days).

3.4. New features

New features were developed based on the original dataset from Kepler that was obtained from MAST to optimize the analysis with *ThetaRay* algorithm. These features were constructed from the original dataset as described below using the phase-folded "Local View" light curves (see, e.g., Shallue and Vanderburg, 2018).

- `global_view` — the original vector of length 2001 or a "Global View" of the TCE that shows the characteristics of the light curve over an entire orbital period. Because of the size limitations of the *ThetaRay*'s system, we performed dimension reduction. We represented groups of 20 columns in the "Global View" by computing the average and standard deviation of those columns. We have a total of 200 new `global_view` features.
- `spline_bkspace` — the breakpoint spacing in time units, used for the best-fit spline. We chose the optimal spacing of spline breakpoints for each light curve by fitting splines with different breakpoint spacings, calculating the Bayesian Information Criterion (BIC, Schwarz, 1978) for each spline, and choosing the breakpoint spacing that minimized the BIC. Below, is a brief
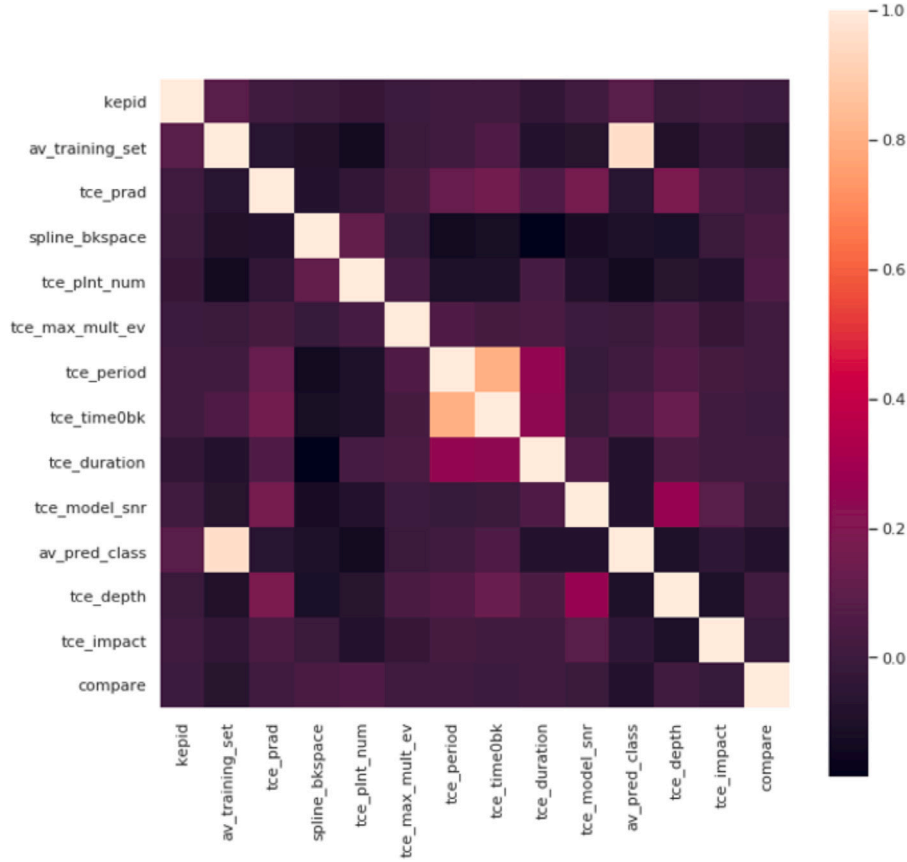
Fig. 5. The 'Heat map' of some of the features (or parameters) used in the *ThetaRay* algorithm. The intensity scale indicates the magnitude of the correlation between the features that facilitates determining the dimensionality of the dataset (see text).

description of the new features that were computed for each TCE "Global View" and "Local View" light curves:

- `loc_mean` — average of the "Local View" light curve.
- `loc_std` — standard deviation of the "Local View" light curve.
- `loc_25%` — 25% percentile of the "Local View" light curve.
- `loc_75%` — 75% percentile of the "Local View" light curve.
- `loc_max` — maximum value of the "Local View" light curve.
- `glob_mean` — average of the original "Global View" light curve.
- `glob_std` — standard deviation of the original "Global View" light curve.
- `glob_25%` — lower percentage of the original "Global View" light curve.
- `glob_75%` — upper percentage of the original "Global View" light curve.
- `glob_max` — maximum value of the original "Global View" light curve.
- `zScore_loc_min` — minimum value of the Z-Score on the "Local View" light curve with window of 10.
- `zScore_loc_max` — maximum value of the Z-Score on the "Local View" light curve with window of 10.
- `zScore_glob_min` — minimum value of the Z-Score on the "Global View" light curve with window of 100.
- `zScore_glob_max` — maximum Z of the-Score on the "Global View" light curve with window of 100.

4. TESS satellite data analysis

4.1. Preprocessing the data

We obtained 10,803 light curves of TCEs produced by the TESS mission from MAST (http://archive.stsci.edu/). Our goal was to use the

same model we built based on Kepler's data, in order to find potential exoplanets (anomalies) in the new data from TESS. In order to use the same models for the two different satellites, we had to convert the TESS data to the same structure as Kepler data. Therefore, we performed additional steps to prepare the light curves to be used as inputs to *ThetaRay*'s system. We generated a set of TFRecord files for the TCEs in TESS data. Each file contains `global_view`, `local_view` and `spline_bkspace` representations like in Kepler data. We also created in python the following data files:

- `global_view` — Vector of length 2001 that shows the characteristics of the light curve over an entire orbital period.
- `local_view` — Vector of length 201 that shows the shape of the transit in detail (phase-folded close-up of the transit event).
- `more_features` — includes

  - `ticid` — TESS ID of the target star.
  - `planetNumber` — TCE number within the target star.
  - `planetRadiusEarthRadii` — has the same meaning as the field of `tce_prad` in Kepler data.
  - `spline_bkspace`, the breakpoint spacing in time units, used for the best-fit spline — same as in Kepler data.
  - `mes` — same meaning as `tce_max_mult_ev` in Kepler data.
  - `orbitalPeriodDays` — same meaning as `tce_period` in Kepler data.
  - `transitEpochBtjd` — same meaning as `tce_time0bk` in Kepler data.
  - `transitDurationHours` — same meaning as `tce_duration` in Kepler data.
  - `transitDepthPpm` — same meaning as `tce_depth` in Kepler Data.

Table 2

Some of the parameters (see text) of exoplanetary candidates identified by the *ThetaRay* system in TESS mission data archive (see text). Column 's' indicates the sector(s).

TIC_ID	# p	s	tce_prad	tce_max_mult_ev	tce_period	tce_time0bk	tce_duration	tce_model	tce_depth	tce_impact
101948569	1	2	3.049010038	11.03339958	19.47240067	1360.109985	3.47104	11.74638202	1645.949951	0.303943992
102195674	1	9	20.62459946	68.38349915	4.378769875	1547.459961	3.8184	67.04560811	30425.69922	0.00999983
117979694	1	5	2.017019987	8.018360138	0.991541028	1438.709961	1.99801	10.88359612	662.7739868	0.557057977
120916706	1	3	5.22453022	10.40649986	0.556737006	1386.170044	0.879121	11.14050062	63679.10156	0.00999983
124573851	1	9	3.842240095	9.629090309	3.948679924	1546.140015	3.42051	11.05563987	754.0189819	0.935890973
164767175	2	3	2.407740116	9.309049606	10.76780033	1393.079956	2.81587	8.523747457	650.6099854	0.202473998
167418903	1	11	10.36499977	20.16550064	21.96240044	1599.300049	1.75464	17.57117062	8278.05957	0.844699979
167603396	1	1–3	2.751130104	8.034460068	14.37919998	1365.140015	6.48194	8.00441099	1444.5	0.269908011
170849515	1	5	10.83209991	16.08620071	1.941280007	1438.150024	1.80662	20.93755124	37131.69922	0.00999983
172464366	1	6	19.09110069	79.50800323	2.921689987	1470.050049	3.17791	77.20788698	17562.40039	0.552250981
178155732	1	1–36	2.372940063	10.47840023	5.971879959	1415.630005	3.12355	12.1434367	316.0220032	0.226411998
178819686	2	10	3.705120087	9.265979767	12.27630043	1572.969971	3.65638	10.82845891	976.2700195	0.603969991
200591694	1	6	4.385819912	8.523739815	13.58699989	1470.150024	2.49368	9.010778093	4702.040039	0.00999983
219345200	1	4	4.915830135	44.37039948	21.9904	1414.579956	4.44366	15.23623284	922.9470215	0.342040002
219379012	1	6	4.340690136	15.25220013	1.546159983	1469.709961	1.34256	17.6102274	1064.26001	0.692296982
219403686	1	6	5.821829796	22.26140022	0.380145997	1468.579956	0.692123	29.63871326	1336.619995	0.679122984
254700590	1	12	2.881239891	7.790110111	11.77639961	1625.609985	4.64722	7.599164669	543.3380127	0.651623011
259377017	3	5	1.372750044	9.26651001	3.359859943	1387.089966	1.40678	8.651269232	1034.959961	0.463200003
260647166	2	11	2.615370035	11.52190018	15.85879993	1606.160034	3.89359	11.53021367	829.5319824	0.00999983
270677759	1	11	9.437470436	14.45300007	9.129110336	1597.199951	4.72452	14.20179783	8185.049805	0.805234015
293844307	1	8	4.286910057	10.61699963	14.76070023	1525.550049	3.56351	10.77202809	1557.839966	0.201021999
296972860	1	9	11.50829983	17.99360085	12.45800018	1549.77002	3.38762	12.5118297	2250.340088	0.990104973
300560295	2	12	5.527339935	7.324260235	6.884880066	1625.630005	2.36212	7.720771781	1999.51001	0.754741013
302971087	1	6	1.856520057	7.757989483	17.5326004	1469.609985	1.81434	6.415764637	1347.23999	0.306959003
303051566	1	1	3.854789972	8.216239929	19.6228	1328.569946	1.49412	6.706231497	569.2410278	0.0374793
306735585	1	1–6	8.496970177	12.65380001	4.816760063	1414.079956	2.29767	11.34648808	5228.529785	0.893122017
307210830	3	11	0.869957983	11.3927002	2.253309965	1598.23999	1.01187	10.71902329	723.9060059	0.501681983
307467401	1	6	4.158410072	44.08229828	9.587329865	1475.709961	6.74873	25.5473955	1375.969971	0.83335799
308994098	1	9	5.175449848	19.70980072	10.51659966	1552.050049	10.7063	18.35474766	991.1049805	0.707704008
309619055	1	9	10.75909996	20.72480011	10.55350018	1604.969971	4.59559	11.93659993	9440.740234	0.894083023
311925584	1	11	14.58570004	19.55500031	3.645989895	1599.189941	1.35795	13.91252537	8853.110352	0.96806401
313676015	1	11	8.788999557	12.18910027	12.95289993	1604.140015	5.25315	11.46735907	3192.100098	0.987146974
322900369	1	1–36	7.504670143	95.60189819	3.126100063	1493.890015	12.7532	92.18038804	5151.629883	0.570958972
335452175	1	11	15.31970024	58.07910156	15.49790001	1601.26001	3.22331	60.38542514	8295.75	0.990104973
355509914	1	1–36	10.77110004	7.868070126	1.738260031	1326.119995	1.29881	8.795540018	18978.40039	0.200622007
365638739	1	6	3.920919896	8.19054985	17.21910095	1470.390015	3.62824	7.832664711	2136.790039	0.301687002
370228465	2	1–2	4.152969837	7.710509777	12.32479954	1357.910034	15.6542	7.614795753	4594.089844	0.0166821
392113518	1	11	5.140649796	11.11190033	4.366240025	1569.849976	5.57279	11.75887721	847.0930176	0.416761011
393474846	1	10	1.801599979	7.123549938	11.05160046	1578.839966	2.60486	6.741978404	305.5039978	0.461097002
396902326	1	9	6.775030136	13.27799988	15.89210027	1545.609985	7.86604	14.09583373	4045.179932	0.987653017
406872253	1	12	2.106899977	7.377220154	0.246232003	1625.170044	0.392131	6.283381761	394.3840027	0.165623993
410214984	3	1–36	4.706439972	12.51220036	8.135899544	1332.349976	3.28862	6.290396653	4277.52002	0.144591004
421951960	1	1–36	3.553849936	8.082489967	11.79030037	1332.48999	1.60633	8.191518961	761.4229736	0.390895009
422655579	1	4	15.61709976	27.75729942	2.903460026	1413.140015	5.02954	45.99784581	4657.879883	0.00999983
423275733	1	8	17.97360039	24.79450035	2.052979946	1518.689941	2.65216	36.4699389	10176.90039	0.745383978
447061717	1	10	2.608789921	21.75650024	9.204919815	1569.719971	3.24093	18.51823286	4082.899902	0.0298279
453767182	1	12	2.725820065	7.798190117	10.76249981	1626.119995	2.93366	7.410331196	13611	0.027590601
455278250	1	8	7.306509972	36.73529816	15.60929966	1521.51001	5.72957	28.64738963	2824.179932	0.82368201

– minImpactParameter — same meaning as `tce_impact` in Kepler data.

TESS data is unlabeled, so `av_training_set` and `av_pred_class` fields do not exist in the TESS data, therefore, we filled these fields with zeros. `tce_model_snr` feature value was calculated from the ratio of `transitDepthPpm` and `transit-DepthPpm_err`.

- `Describe files` — includes count, mean, std, min, max, 25% percentile, median (50%), 75% percentile. These quantities were computed on each original data row from the `global_view` and `local_view` files and on each scaling row of these files.

Following the generation of the dataset in the form of Comma Separated Values (CSVs), we applied the same manipulation on `global_view`, as in Kepler data, in order to reduce the dimensions, and used the analogous 424 features produced from TESS data as in Kepler data, for the analysis on *ThetaRay*'s system. Following this step, we applied the `Detection` algorithm on TESS data according to the saved model from Kepler and used the results for classification and mapping of TESS light curve TCEs data.

4.2. Results: Transiting exoplanet detection

The first results of the *ThetaRay* algorithm that was applied to TESS data TCEs in December 2019 produced around 90 preliminary targets that were further manually vetted removing obvious non-planets and 12 artifact cases due to momentum dumps of the TESS satellite (see, e.g., Vanderburg et al., 2019), and thus reducing the number of targets for analysis by about a factor of two. "Local View" light-curves were used together with planetary candidate parameters to vet the algorithm's output. In the manual vetting the physical parameters, such as non-typical "Local View" light curves (i.e., v-shapes, and other non-planetary periodic features), extremely large planetary radius ($r \gtrsim 20R_{Earth}$), and very low signal-to-noise were used. The parameters for the remaining 48 cases identified as EPCs by the *ThetaRay* system form the TESS database of 10,803 TCE's are given in Table 2, and the disposition of each identified EPC is provided in Table 3. It is evident that three identified cases are EPCs, while the remaining cases are known exoplanets, planetary candidates, and signals due to other effects, such as off-target light-curves, and scattered light. Cross-checking our results with the TESS planet candidate list https://tev.mit.edu/data/collection/193/, matches 19 cases in our table of planet candidates and confirmed planes as indicated in Table 3. While these

19 cases are not new EPCs, the overlap with the MIT planet candidate list improves confidence in the performance of the *ThetaRay* algorithm in identifying EPCs.

In Fig. 6 we show the "Local View" light curves of eight selected cases for the objects identified using the *ThetaRay* algorithm as potential EPCs. The TESS input catalog ID number (TIC_ID), along with several parameters (`tce_prad, tce_period, tce_depth` defined in Section 3.2) for the identified EPC are indicated on each panel. The eight panels in Fig. 6 show typical eclipsing exoplanetary light curve temporal shape structure. In particular TIC_ID: 178155732 was identified by the algorithm, and is a known planet HR 858 c, TIC_ID: 422655579 is a known planet WASP-71 b, and TIC_ID: 308994098 is a known planet candidate TOI 790. While these objects are not new EPCs, their identification provides validation of the ML algorithm. The objects TIC_ID: 219345200, 219403686, 306735585, 421951960, shown in Fig. 6 are possible new EPCs identified by the algorithm (with some reservation with respect to TIC_ID: 306735585 indicated in Table 3).

We note that Table 2 contains only three case with planetary radius (`tce_prad`) or $r_p < 2R_{Earth}$ (TIC_ID 307210830 known planet L98-59 b, TIC_ID 259377017 known planet TOI 270 b, and TIC_ID 305751503 new EPC), and a total of 17 EPCs identified with $r_p < 3R_{Earth}$. Another 13 identified EPCs were similar in size or larger than Jupiter with $r_p \geq 10R_{Earth}$, among them 6 were previously identified planets or TOIs, 3 are artifacts due instrumental effects, and 3 are new EPCs. We find the following properties of the data in Table 2

- The orbital periods (`tce_period`) of the identified EPCs range from 0.25d to just under 22d.
- The transit depth (`tce_depth`) varied by about an order of magnitude in the range $305 - 63679$ ppm with the signal-to-noise in the range $\sim 5 - 92$ ppm.
- The impact parameter was in the range $\sim 0.01 - 0.99$.
- The duration of the transits (`tce_duration`) was in the range $0.392 - 15.65$ h.
- In 8 cases the identified EPCs suggest multiple planetary systems with 2 and 3 planets.
- Three new EPCs were identified.

5. Discussion and conclusions

The TESS satellite provides observations of a large number (200,000) of stellar light curves with high photometric precision over the whole sky, divided in observing sectors, with the aim of detecting transiting Earth-sized planets. The stellar objects were selected to represent the brightest and closest to our solar system. The large dataset of nearly 27 GB per day is then processed in the science data pipeline providing nearly 11,000 TCE's as of the time of writing this paper. Further analysis of the TCEs is required to find confirmed examples of exoplanets, or exoplanetary candidates for more in-depth processing. However, evidently this formidable data analysis task is difficult, if not impossible to carry out manually. A feasible approach for the TESS data analysis is based on automated identification techniques that were developed recently, customized for transiting exoplanetary candidates identification, EPCs, utilizing AI/ML methods based on DL neural networks machine learning methods combined with anomaly identification methods reported the present study. These EPCs could be than vetted further with targeted observations and data analysis.

In this study we apply a novel algorithm developed by *ThetaRay, Inc.* for cybersecurity and anomaly identification in financial systems to help detected exoplanets in TESS data. This is the first attempt to do this task in an unsupervised or semi-supervised machine learning. The advantage of this AI/ML system over other unsupervised ML methods is the combination of several algorithms, as described in this paper (Section 2), and the direct application to any large dataset that contain possibly small number of target data points ('anomalies'). We apply the system to TESS observations of TCE's in search of transiting exoplanet

Table 3

The disposition of exoplanetary candidates identified by the *ThetaRay* system in TESS mission data archive.

TIC_ID	Disposition
101948569	Known planet candidate TOI 240
102195674	Known planet candidate TOI 668
117979694	Known planet candidate TOI 424/off target
120916706	Known planet candidate TOI 263
124573851	Known planet candidate TOI 669
164767175	Known planet candidate TOI 266
167418903	Known planet candidate (detected at the wrong 2x period) TOI 746
167603396	Transit/possibly hartbeat star[a]
170849515	Known planet candidate TOI 555
172464366	Known planet candidate (confirmed to be an eclipsing binary) TOI 478
178155732	Known planet HR 858 c
178819686	Known planet candidate TOI 763
200591694	Off-target
219345200	Possible new planet candidate
219379012	Off-target
219403686	Possible new planet candidate
254700590	Possibly single transit?
259377017	Known planet TOI 270
260647166	Known planet candidate TOI 1233
270677759	Off-target
293844307	Off-target
296972860	Signal apparently due to an asteroid in the solar system passing near the target star affecting the background light subtraction[b]
300560295	Off-target
302971087	Atypical transit shape
303051566	Possibly single transit?
306735585	Possible planet but ghost diagnostic is high[c]
307210830	Known planet L 98-59
307467401	Scattered light event[d]
308994098	Known planet candidate TOI 790
309619055	Off-target
311925584	Significant odd/even transit depth difference
313676015	Single transit/off-target
322900369	Clear eclipsing binary with tidal distortion right after eclipse
335452175	Off-target
355509914	Known planet candidate TOI 202/Possibly off-target[c]
365638739	Single transit/off-target
370228465	Possibly single transit?
392113518	Secondary eclipse/off-target
393474846	Stellar variability/not transit-shaped light curve
396902326	Off-target
406872253	Off-target
410214984	Signal due to known planet DS Tuc Ab (on nearby star)
421951960	Possible new planet candidate
422655579	Known planet WASP-71 b
423275733	Known planet WASP 142 b
447061717	Known planet candidate TOI 1231
453767182	Off target
455278250	Scattered light

[a] See, e.g., Shporer et al. (2016).
[b] See, http://vo.imcce.fr/webservices/skybot/?forms=conesearch.
[c] See, Bryson et al. (2013) and Jenkins et al. (2016) for the analysis method description.
[d] See, https://docs.lightkurve.org/tutorials/2-creating-light-curves/2-3-removing-scattered-light-using-regressioncorrector.html Dalba et al. (2020), Hattori et al. (2021).

signatures in the large TCE dataset. For the training set of the ML algorithm, we used the Kepler exoplanet TCE's validated with confirmed exoplanet dataset. By applying the trained *ThetaRay* algorithm to TESS TCE's we identified 48 targets for further analysis in wide range of sizes from below Earth's radius to super-Jupiter's radii, and planetary periods ranging from 0.38d to ~20d. Further manual vetting resulted in the identification of three new exoplanetary candidates (see Table 3), with the rest identified as known planets, known planatary candidates, and observational artifacts. We demonstrate that the combination of DL neural networks with anomaly identification mathematical techniques provide an efficient AI/ML algorithm for the rapid automated search of transiting exoplanet candidates light curves in a semi-supervised
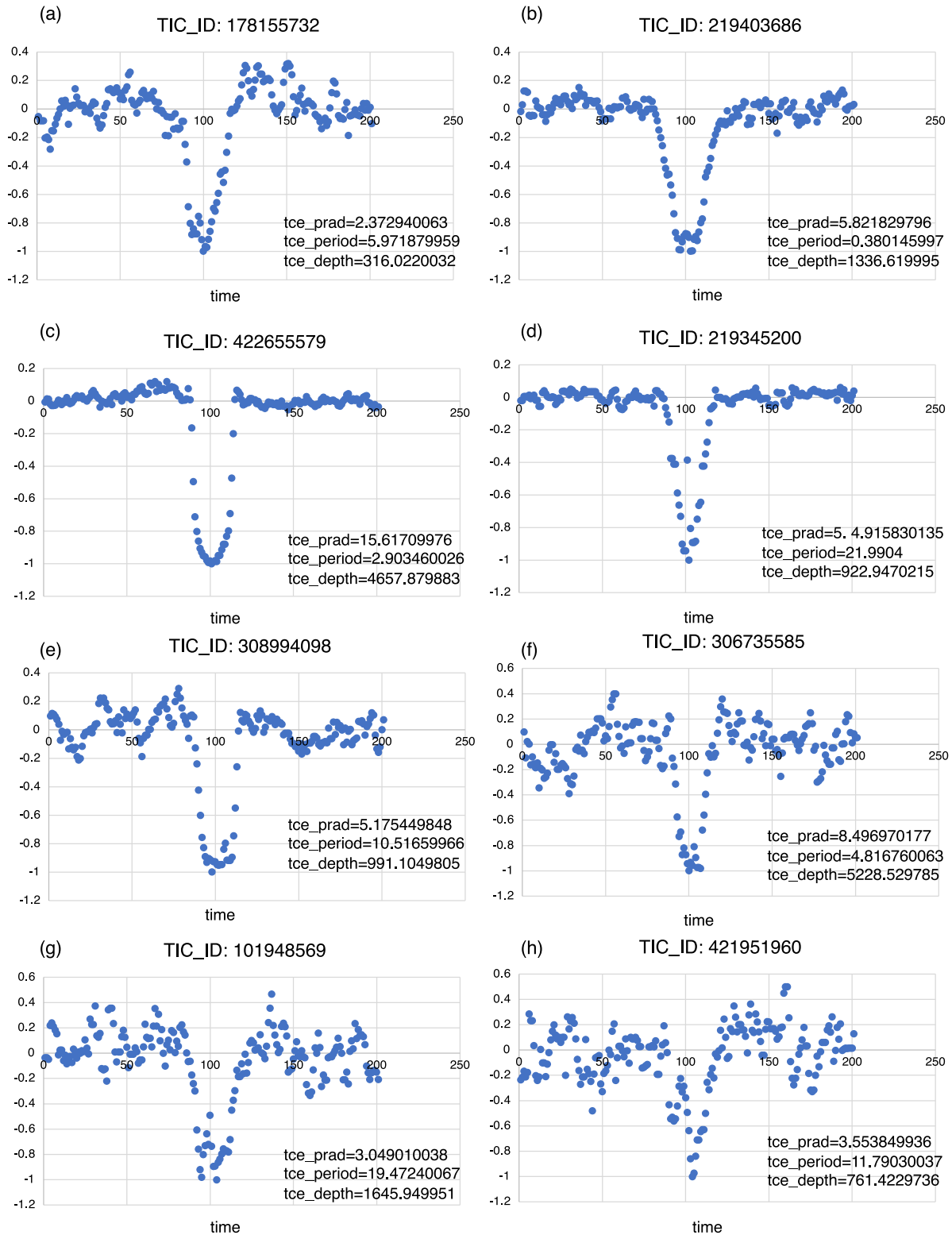
Fig. 6. "Local view" normalized phase-folded light-curves of six objects selected from Table 2 representing different types of detections by the ThetaRay algorithm. The parameters tce_prad the radius in terms of R_{Earth}, tce_period in days, tce_depth in ppm, indicated on the corresponding panels. The typical eclipsing exoplanetary light curve temporal shape structure is evident. (a) TIC_ID: 178155732: known planet HR 858 c, (b) TIC_ID: 219403686: possible new planet candidate, (c) TIC_ID: 422655579: known planet WASP-71 b, (d) TIC_ID: 219345200: possible new planet candidate, (e) TIC_ID: 308994098 is a known planet candidate TOI 790, (f) TIC_ID: 308994098: known planet candidate TOI 790, (g) TIC_ID: 306735585: possible new planet candidate (but see comment in Table 3), (h) TIC_ID: 421951960: possible new planet candidate.

and unsupervised approach. Although, we find that we need to apply manual vetting to reduce the number of obvious false-positives or known cases, the total number of EPCs identifications is manageable for secondary manual vetting of the relatively small number of light-curves, and this approach provides the desired classification of the results. In future applications, the *ThetaRay* algorithm could be further optimized for transiting exoplanets light-curves classification helped the identification of EPCs, by including, for example, informed ML steps, potentially reducing further the false-positive rate in this application and providing a new tool for analyzing TESS data.

Declaration of competing interest

Leon Ofman has consulted *ThetaRay* on the subject of this article. Amir Averbuch, Adi Shliselberg, Idan Benaun, David Segev and Aron Rissman were employed by *ThetaRay* and Averbuch is also the founder.

Acknowledgments

References

Armstrong, D.J., Gamper, J., Damoulas, T., 2020. Exoplanet validation with machine learning: 50 new validated Kepler planets. ArXiv E-Prints arXiv:2008.10516.

Bérard, P., Besson, G., Gallot, S., 1994. Embedding Riemannian manifolds by their heat kernel. Geom. Funct. Anal. 4 (4), 373–398.

Bermanis, A., Salhov, M., Averbuch, A., 2021. Geometric component analysis and its applications to data analysis. Appl. Comput. Harmon. Anal.

Borucki, W.J., Koch, D., Basri, G., Batalha, N., Brown, T., Caldwell, D., Caldwell, J., Christensen-Dalsgaard, J.r., Cochran, W.D., DeVore, E., Dunham, E.W., Dupree, A.K., Gautier, T.N., Geary, J.C., Gilliland, R., Gould, A., Howell, S.B., Jenkins, J.M., Kondo, Y., Latham, D.W., Marcy, G.W., Meibom, S.r., Kjeldsen, H., Lissauer, J.J., Monet, D.G., Morrison, D., Sasselov, D., Tarter, J., Boss, A., Brownlee, D., Owen, T., Buzasi, D., Charbonneau, D., Doyle, L., Fortney, J., Ford, E.B., Holman, M.J., Seager, S., Steffen, J.H., Welsh, W.F., Rowe, J., Anderson, H., Buchhave, L., Ciardi, D., Walkowicz, L., Sherry, W., Horch, E., Isaacson, H., Everett, M.E., Fischer, D., Torres, G., Johnson, J.A., Endl, M., MacQueen, P., Bryson, S.T., Dotson, J., Haas, M., Kolodziejczak, J., Van Cleve, J., Chandrasekaran, H., Twicken, J.D., Quintana, E.V., Clarke, B.D., Allen, C., Li, J., Wu, H., Tenenbaum, P., Verner, E., Bruhweiler, F., Barnes, J., Prsa, A., 2010. Kepler planet-detection mission: Introduction and first results. Science 327 (5968), 977. http://dx.doi.org/10.1126/science.1185402.

Brown, T.M., Latham, D.W., Everett, M.E., Esquerdo, G.A., 2011. Kepler input catalog: Photometric calibration and stellar classification. Astron. J. 142 (4), 112. http://dx.doi.org/10.1088/0004-6256/142/4/112, arXiv:1102.0342.

Bryson, S.T., Jenkins, J.M., Gilliland, R.L., Twicken, J.D., Clarke, B., Rowe, J., Caldwell, D., Batalha, N., Mullally, F., Haas, M.R., Tenenbaum, P., 2013. Identification of background false positives from Kepler data. PASP 125 (930), 889. http://dx.doi.org/10.1086/671767, arXiv:1303.0052.

Catanzarite, J.H., 2015. Autovetter Planet Candidate Catalog for Q1-Q17 Data Release 24. KSCI-19091-001, NASA Ames Research Center, Moffett Field, CA.

Christiansen, J.L., Jenkins, J.M., Caldwell, D.A., Burke, C.J., Tenenbaum, P., Seader, S., Thompson, S.E., Barclay, T.S., Clarke, B.D., Li, J., Smith, J.C., Stumpe, M.C., Twicken, J.D., Cleve, J.V., 2012. The derivation, properties, and value of Kepler's combined differential photometric precision. Publ. Astron. Soc. Pac. 124 (922), 1279–1287. http://dx.doi.org/10.1086/668847.

Coifman, R.R., Lafon, S., 2006a. Diffusion maps. Appl. Comput. Harmon. Anal. 21 (1), 5–30.

Coifman, R.R., Lafon, S., 2006b. Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions. Appl. Comput. Harmon. Anal. 21 (1), 31–52.

Coughlin, J.L., Mullally, F., Thompson, S.E., Rowe, J.F., Burke, C.J., Latham, D.W., Batalha, N.M., Ofir, A., Quarles, B.L., Henze, C.E., Wolfgang, A., Caldwell, D.A., Bryson, S.T., Shporer, A., Catanzarite, J., Akeson, R., Barclay, T., Borucki, W.J., Boyajian, T.S., Campbell, J.R., Christiansen, J.L., Girouard, F.R., Haas, M.R., Howell, S.B., Huber, D., Jenkins, J.M., Li, J., Patil-Sabale, A., Quintana, E.V., Ramirez, S., Seader, S., Smith, J.C., Tenenbaum, P., Twicken, J.D., Zamudio, K.A., 2016. Planetary candidates observed by Kepler. VII. The first fully uniform catalog based on the entire 48-month data set (Q1-q17 DR24). Astrophys. J. Suppl. 224 (1), 12. http://dx.doi.org/10.3847/0067-0049/224/1/12, arXiv:1512.06149.

Dalba, P.A., Gupta, A.F., Rodriguez, J.E., Dragomir, D., Huang, C.X., Kane, S.R., Quinn, S.N., Bieryla, A., Esquerdo, G.A., Fulton, B.J., Scarsdale, N., Batalha, N.M., Beard, C., Behmard, A., Chontos, A., Crossfield, I.J., Dressing, C.D., Giacalone, S., Hill, M.L., Hirsch, L.A., Howard, A.W., Huber, D., Isaacson, H., Kosiarek, M., Lubin, J., Mayo, A.W., Mocnik, T., Murphy, J.M.A., Petigura, E.A., Robertson, P., Rosenthal, L.J., Roy, A., Rubenzahl, R.A., Zandt, J.V., Weiss, L.M., Knudstrup, E., Andersen, M.F., Grundahl, F., Yao, X., Pepper, J., Villanueva, S., Ciardi, D.R., Cloutier, R., Jacobs, T.L., Kristiansen, M.H., LaCourse, D.M., Lendl, M., Osborn, H.P., Palle, E., Stassun, K.G., Stevens, D.J., Ricker, G.R., Vanderspek, R., Latham, D.W., Seager, S., Winn, J.N., Jenkins, J.M., Caldwell, D.A., Daylan, T., Fong, W., Goeke, R.F., Rose, M.E., Rowden, P., Schlieder, J.E., Smith, J.C., Vanderburg, A., 2020. The TESS–keck survey. I. A warm sub-saturn-mass planet and a caution about stray light in TESS cameras. Astron. J. 159 (5), 241. http://dx.doi.org/10.3847/1538-3881/ab84e3.

Dattilo, A., Vanderburg, A., Shallue, C.J., Mayo, A.W., Berlind, P., Bieryla, A., Calkins, M.L., Esquerdo, G.A., Everett, M.E., Howell, S.B., Latham, D.W., Scott, N.J., Yu, L., 2019. Identifying exoplanets with deep learning. II. Two new super-earths uncovered by a neural network in K2 data. Astron. J. 157 (5), 169. http://dx.doi.org/10.3847/1538-3881/ab0e12, arXiv:1903.10507.

Golovin, D., Solnil, B., Moitra, S., Kochanski, G., Karro, J., D., S., 2017. Google Vizier: A Service for Black-Box Optimization. ACM, p. 1487, ISBN: 978-1-4503-4887-4/17/08. http://dx.doi.org/10.1145/3097983.3098043.

Gu, M., Eisenstat, S.C., 1996. Efficient algorithms for computing a strong rank-revealing QR factorization. SIAM J. Sci. Comput. 17 (4), 848–869.

Hattori, S., Foreman-Mackey, D., Hogg, D.W., Montet, B.T., Angus, R., Pritchard, T., Curtis, J.L., Schölkopf, B., 2021. The unpopular package: a data-driven approach to de-trend TESS full frame image light curves. arXiv:2106.15063.

Jenkins, J.M., Caldwell, D.A., Chandrasekaran, H., Twicken, J.D., Bryson, S.T., Quintana, E.V., Clarke, B.D., Li, J., Allen, C., Tenenbaum, P., Wu, H., Klaus, T.C., Cleve, J.V., Dotson, J.A., Haas, M.R., Gilliland, R.L., Koch, D.G., Borucki, W.J., 2010a. Initial characteristics of kepler long cadence data for detecting transiting planets. Astrophys. J. Lett. 713 (2), L120–L125. http://dx.doi.org/10.1088/2041-8205/713/2/l120.

Jenkins, J.M., Caldwell, D.A., Chandrasekaran, H., Twicken, J.D., Bryson, S.T., Quintana, E.V., Clarke, B.D., Li, J., Allen, C., Tenenbaum, P., Wu, H., Klaus, T.C., Middour, C.K., Cote, M.T., McCauliff, S., Girouard, F.R., Gunter, J.P., Wohler, B., Sommers, J., Hall, J.R., Uddin, A., Wu, M.S., Bhavsar, P.A., Cleve, J.V., Pletcher, D.L., Dotson, J.A., Haas, M.R., Gilliland, R.L., Koch, D.G., Borucki, W.J., 2010b. Overview of the Kepler science processing pipeline. Astrophys. J. Lett. 713 (2), L87–L91. http://dx.doi.org/10.1088/2041-8205/713/2/l87.

Jenkins, J.M., Twicken, J.D., McCauliff, S., Campbell, J., Sanderfer, D., Lung, D., Mansouri-Samani, M., Girouard, F., Tenenbaum, P., Klaus, T., Smith, J.C., Caldwell, D.A., Chacon, A., Henze, C., Heiges, C., Latham, D.W., Morgan, E., Swade, D., Rinehart, S., Vanderspek, R., 2016. The TESS science processing operations center. In: Chiozzi, G., Guzman, J.C. (Eds.), Software and Cyberinfrastructure for Astronomy IV. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 9913, p. 99133E. http://dx.doi.org/10.1117/12.2233418.

Johnson, W.B., Lindenstrauss, J., 1984. Extensions of Lipschitz mappings into a Hilbert space. Contemp. Math. 26 (189–206), 1.

Koch, D.G., Borucki, W.J., Basri, G., Batalha, N.M., Brown, T.M., Caldwell, D., rgen Christensen-Dalsgaard, J., Cochran, W.D., DeVore, E., Dunham, E.W., Gautier, T.N., Geary, J.C., Gilliland, R.L., Gould, A., Jenkins, J., Kondo, Y., Latham, D.W., Lissauer, J.J., Marcy, G., Monet, D., Sasselov, D., Boss, A., Brownlee, D., Caldwell, J., Dupree, A.K., Howell, S.B., Kjeldsen, H., ren Meibom, S., Morrison, D., Owen, T., Reitsema, H., Tarter, J., Bryson, S., Dotson, J.L., Gazis, P., Haas, M.R., Kolodziejczak, J., Rowe, J.F., Cleve, J.E.V., Allen, C., Chandrasekaran, H., Clarke, B.D., Li, J., Quintana, E.V., Tenenbaum, P., Twicken, J.D., Wu, H., 2010. Kepler mission design, realized photometric performance, and early science. Astrophys. J. Lett. 713 (2), L79–L86. http://dx.doi.org/10.1088/2041-8205/713/2/l79.

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation forest. In: 2008 Eighth Ieee International Conference on Data Mining. IEEE, pp. 413–422.

Mandel, K., Agol, E., 2002. Analytic light curves for planetary transit searches. Astrophys. J. Lett. 580 (2), L171–L175. http://dx.doi.org/10.1086/345520, arXiv:astro-ph/0210099.

Osborn, H., Ansdell, M., Ioannou, Y., Sasdelli, M., Angerhausen, D., Caldwell, D., Jenkins, J., Räissi, C., Smith, J., 2020. Rapid classification of TESS planet candidates with convolutional neural networks. Astron. Astrophys. 633, A53. http://dx.doi.org/10.1051/0004-6361/201935345, arXiv:1902.08544.

Pan, C.-T., 2000. On the existence and computation of rank-revealing LU factorizations. Linear Algebra Appl. 316 (1–3), 199–222.

Ricker, G.R., Winn, J.N., Vanderspek, R., Latham, D.W., Bakos, G.A., Bean, J.L., Berta-Thompson, Z.K., Brown, T.M., Buchhave, L., Butler, N.R., Butler, R.P., Chaplin, W.J., Charbonneau, D., Christensen-Dalsgaard, J.r., Clampin, M., Deming, D., Doty, J., De Lee, N., Dressing, C., Dunham, E., Endl, M., Fressin, F., Ge, J., Henning, T., Holman, M.J., Howard, A.W., Ida, S., Jenkins, J., Jernigan, G., Johnson, J.A., Kaltenegger, L., Kawai, N., Kjeldsen, H., Laughlin, G., Levine, A.M., Lin, D., Lissauer, J.J., MacQueen, P., Marcy, G., McCullough, P., Morton, T.D., Narita, N., Paegert, M., Palle, E., Pepe, F., Pepper, J., Quirrenbach, A., Rinehart, S., Sasselov, D., Sato, B., Seager, S., Sozzetti, A., Stassun, K.G., Sullivan, P., Szentgyorgyi, A., Torres, G., Udry, S., Villasenor, J., 2014. Transiting exoplanet survey satellite (TESS). In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 9143, 914320. http://dx.doi.org/10.1117/12.2063489.

Schwarz, G., 1978. Estimating the dimension of a model. Ann. Statist. 6 (2), 461–464. http://dx.doi.org/10.1214/aos/1176344136.

Shabat, G., Segev, D., Averbuch, A., 2018a. Uncovering unknown unknowns in financial services big data by unsupervised methodologies: Present and future trends. In: Proceedings of Machine Learning Research, KDD 2017 Workshop on Anomaly Detection in Finance, pp. 8–19.

Shabat, G., Shmueli, Y., Aizenbud, Y., Averbuch, A., 2018b. Randomized LU decomposition. Appl. Comput. Harmon. Anal. 44, 246–272.

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. http://dx.doi.org/10.3847/1538-3881/aa9e09, arXiv:1712.05044.

Shporer, A., Fuller, J., Isaacson, H., Hambleton, K., Thompson, S.E., Prša, A., Kurtz, D.W., Howard, A.W., O'Leary, R.M., 2016. Radial velocity monitoring of Kepler heartbeat stars. Astrophys. J. 829 (1), 34. http://dx.doi.org/10.3847/0004-637X/829/1/34, arXiv:1606.02723.

Vanderburg, A., Huang, C.X., Rodriguez, J.E., Becker, J.C., Ricker, G.R., Vanderspek, R.K., Latham, D.W., Seager, S., Winn, J.N., Jenkins, J.M., Addison, B., Bieryla, A., Briceño, C., Bowler, B.P., Brown, T.M., Burke, C.J., Burt, J.A., Caldwell, D.A., Clark, J.T., Crossfield, I., Dittmann, J.A., Dynes, S., Fulton, B.J., Guerrero, N., Harbeck, D., Horner, J., Kane, S.R., Kielkopf, J., Kraus, A.L., Kreidberg, L., Law, N., Mann, A.W., Mengel, M.W., Morton, T.D., Okumura, J., Pearce, L.A., Plavchan, P., Quinn, S.N., Rabus, M., Rose, M.E., Rowden, P., Shporer, A., Siverd, R.J., Smith, J.C., Stassun, K., Tinney, C., Wittenmyer, R., Wright, D.J., Zhang, H., Zhou, G., Ziegler, C.A., 2019. TESS spots a compact system of super-earths around the naked-eye star HR 858. Astrophys. J. Lett. 881 (1), L19. http://dx.doi.org/10.3847/2041-8213/ab322d, arXiv:1905.05193.

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., Vanderspek, R.K., Latham, D.W., Seager, S., Dittmann, J., Doty, J.P., Glidden, A., Quinn, S.N., 2019. Identifying exoplanets with deep learning. III. Automated triage and vetting of TESS candidates. Astron. J. 158 (1), 25. http://dx.doi.org/10.3847/1538-3881/ab21d6, arXiv:1904.02726.

Zucker, S., Giryes, R., 2018. Shallow transits—deep learning. I. Feasibility study of deep learning to detect periodic transits of exoplanets. Astron. J. 155 (4), 147. http://dx.doi.org/10.3847/1538-3881/aaae05, arXiv:1711.03163.