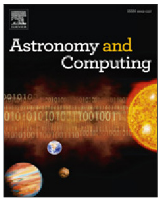
Astronomy and Computing 30 (2020) 100335

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom



Full length article

CEESA meets machine learning: A Constant Elasticity Earth Similarity Approach to habitability and classification of exoplanets

S. Basak [a], S. Saha [b], A. Mathur [c], K. Bora [d,*], S. Makhija [d], M. Safonova [e], S. Agrawal [f]

[a] Department of Computer Science and Engineering, University of Texas at Arlington, USA
[b] Department of Computer Science and Information Systems and APPCAIR, BITS Pilani K.K. Birla Goa Campus, India
[c] Department of Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India
[d] Department of Information Science and Engineering, and Center for AstroInformatics Modeling and Simulation (CAMS), PES University South Campus, Bangalore, India
[e] Indian Institute of Astrophysics, Bangalore, India
[f] Department of Computer Science and Engineering, DY Patil University, Pune, India

ARTICLE INFO

Article history:
Received 10 March 2019
Accepted 16 October 2019
Available online 14 November 2019

Keywords:
Machine learning
Exoplanets
Particle Swarm Optimization
Astroinformatics
Habitability

ABSTRACT

We examine the existing metrics of habitability and classification schemes of extrasolar planets and provide an exposition of the use of computational intelligence techniques to estimate habitability and to automate the process of classification of exoplanets. Exoplanetary habitability is a challenging problem in Astroinformatics, an emerging area in computational astronomy. The paper introduces a new constant elasticity habitability metric, the 'Constant Elasticity Earth Similarity Approach (CEESA)', to address the shortcoming of previous metrics. The proposed metric incorporates eccentricity as one of the component features to estimate the potential habitability of extrasolar planets. CEESA is a novel optimization model and computes habitability scores within the framework of a constrained optimization problem solved by metaheuristic method, mitigating the complexity and curvature violation issues in the process. The metaheuristic method, developed in the paper to solve the constrained optimization problem, is a 'derivative-free' optimization method, scope of which is promising beyond the current work. Habitabilty scores, such as CDHS (Bora et al., 2016), are recomputed with the imputed eccentricity values by the method developed in the paper and cross-matched with CEESA scores for validation. The paper also proposes fuzzy neural network-based approach to accomplish classification of exoplanets. Predicted class labels here are independent of CEESA, and are further validated by cross-matching them with the habitability scores computed by CEESA. We conclude by demonstrating the convergence between two proposed approaches, Earth-similarity approach (CEESA) and prediction of habitability labels (classification approach). The convergence between the two approaches establish the efficacy of CEESA in finding potentially habitable planets.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

The rate of discovery of extrasolar planets (exoplanets) is rapidly increasing. The idea that planets other than Earth can possibly harbor life has intrigued and captured human imagination for centuries. Recently, thousands of planets were discovered in our Galaxy alone with the inference that stars with planets are a usual occurrence, and the estimates are that there are at least as many planets in the Milky Way as stars, or even many more (including the free-floating planets; e.g. Strigari et al., 2012; Cassan et al., 2012; van Elteren et al., 2019). Led by the NASA Kepler Mission (Batalha, 2014), planetary searches yielded nearly 7000 confirmed and yet to be confirmed exoplanets (at time of writing). The discovery and characterization of exoplanets require both extremely accurate instrumentation and sophisticated statistical methods in order to extract the weak planetary signals. Their detailed modeling for obtaining the orbital or atmospheric properties is even more challenging. Inferring the properties of underlying planet populations from biased or incomplete samples is another challenge. But characterization of exoplanets is important to judge their habitability — the measure of how probable is the potential of life on a planet. This question is of extreme interest and importance to the humanity because the discovery of even the primitive life on another world will have a profound impact on our civilization. The theoretical work in this regard expanded from the concept of a stellar habitable zone (HZ) to the idea of a Galactic HZ (Gonzalez et al., 2001) and, recently, to the Universe HZ in the context of evaluating which galaxies may be more habitable (Dayal et al., 2015). But the original

[*] Corresponding author.
E-mail address: k_bora@pes.edu (K. Bora).

https://doi.org/10.1016/j.ascom.2019.100335
2213-1337/© 2019 Elsevier B.V. All rights reserved.

question of which of thousands detected planets are habitable, or potentially habitable, is not yet answered. To answer that, we need to understand how different planetary parameters, such as planet's orbital and physical properties, or even host star's physical properties, combine to provide habitable conditions.

Given the increasing rate of discovery of exoplanets (especially with the scheduled launch of the James Webb Space Telescope in 2019), it can be expected that the amount of data samples of exoplanets will reach the scale of a big-data problem (much like the volume of samples collected by the SDSS,[1] which is terabytes in size). In this context, it is important to explore the current classification schemes and to devise methods which can automatically discover meaningful patterns in data and classify them. Since not one single parameter can suffice as the sole criteria for habitability, we explore methods which take into consideration multiple observable characteristics of exoplanets. For example, presence of water may increase the likelihood of an exoplanet to be potentially habitable (Irwin et al., 2014). If a planet resides in the HZ, it is considered to be potentially habitable since the atmospheric conditions in these zones are more likely to support life (Kaltenegger et al., 2011; Méndez, 2011). However, in either case, the habitability cannot be affirmed until other parameters such as planet's orbital and physical properties are collectively considered.

We developed a method which does not require target class labels but finds an optimal convex combination of the observables — *Earth similarity score*. With currently 3875 confirmed and about 3000 unconfirmed discoveries,[2] the amount of accumulated data is rich, and the challenge in determining the potentially habitable candidates lies in the selection of parameters of higher priority. This issue was first addressed in Schulze-Makuch et al. (2011) who formulated two indices, the Planetary Habitability Index (PHI) and the Earth Similarity Index (ESI). To account for the biology related features, another parameter was introduced — the Biological Complexity Index (BCI) (Irwin et al., 2014). Here, we briefly describe the mathematical forms of these parameters.

Earth similarity index (ESI). ESI was designed to determine the exoplanet similarity to Earth (Schulze-Makuch et al., 2011), since we know that life flourishes in Earth-like conditions. ESI range is 0 (no similarity) to 1 (ESI value of the Earth). A planetary body with an ESI over 0.8 is considered to be Earth-like. It was defined in the form

$$ESI_x = \left(1 - \left|\frac{x - x_0}{x + x_0}\right|\right)^w , \qquad (1)$$

with ESI_x being the ESI value of a planet for x property, x_0 the Earth's value for that property, and w the weighting component for adjusting the sensitivity of the scale. Four parameters: surface temperature T_s, density D, escape velocity V_e and radius R, are used to determine the total ESI, through calculating separately the interior ESI_i (from radius and density), and surface ESI_s (from escape velocity and surface temperature). Finally, the total ESI of a planet is calculated by taking the geometric mean of ESI_i and ESI_s. However, ESI in this form (1) only describes the similarity of a planet to the Earth. It does not define the habitability. For example, it is relatively high for the Moon — about 0.5.

Planetary habitability index (PHI). For a quantitative measure of the ability of a planet to develop and sustain life, Schulze-Makuch et al. (2011) defined the PHI index,

$$PHI = (S \cdot E \cdot C \cdot L)^{1/4} , \qquad (2)$$

where S is a substrate, E — available energy, C — appropriate chemistry and L — liquid medium. The PHI value of each parameter is divided by the maximum PHI to normalize the scale to between 0 to 1. However, the PHI parameters are difficult to measure, and it may have missed some other properties that are necessary for determining planet's present habitability. For example, Safonova et al. (2016) proposed to complement the PHI with the age of the planet (see their Eq. 6).

Biological complexity index (BCI). Yet another habitability index was introduced by the same group (Irwin et al., 2014) as an extension of the PHI, with inclusion of geophysical complexity G, temperature T and planetary age A,

$$BCI = (S \cdot E \cdot T \cdot G \cdot A)^{1/5} . \qquad (3)$$

which is then normalized to the maximum BCI value in the set to produce the scale from 0 to 1. Yet, Venus has BCI of zero and Enceladus the BCI of 0.17, while Gliese 581c has the highest BCI of any exoplanet, even higher than the Earth. However, this planet has more of a Venus-like environment being very close to its star. In addition, this index was mainly oriented at assessing the probability of finding a complex (evolved) life on a planetary body.

The standard conservative definition of a habitable planet is applied for planets residing in the classical HZ: a region where liquid water can exist on the surface (Huang, 1959; Kasting, 1993). However, it is possible for a planet to be a good candidate for habitability even outside the classical HZ, or even without a host (Stevenson, 1999; Irwin and Schulze-Makuch, 2011; Heller and Armstrong, 2014). Also, our Moon is within the HZ but clearly is not potentially habitable for our kind of life. Though observational efforts are concentrating on the search for Earth's twin (i.e. the planet with $ESI = 1$), it is quite possible that even with ESI close to 1, a planet is not potentially habitable. Recent 'best bet' for a life-supporting planet, Gliese 832c with $ESI = 0.81$ (Wittenmyer et al., 2014), was found more likely to be a super-Venus and is, probably, tidally locked with its star.

Cobb–douglas habitability production function (CD-HPF). We believe in the probabilistic measure of habitability, in contrast to the classical binary definition of being or not in the HZ. This requires ranking exoplanets in a range of habitability potential by optimizing the habitability production function. Thus, we have introduced a Cobb–Douglas Habitability Score (an offspring of a constrained optimization problem), by using measured and estimated planetary parameters (Bora et al., 2016). The goal was to determine the likelihood of an exoplanet to be (potentially) habitable by comparing its habitability score (CDHS) with the Earth's. The general form of the Cobb–Douglas production function CD-HPF is

$$Y = k \prod_{i=1\ldots n} x_i^{\alpha_i} , \qquad (4)$$

where k is a constant assumed to be 1, Y is the habitability score, i.e. output, x_i are the planetary parameters (or factors), and α_i the elasticity coefficients, determined by the solution to the optimization problem. The sum of α_i (which can be ≤ 1, $= 1$, or ≥ 1) determines returns to scale conditions in the CD-HPF (Saha et al., 2016). We have shown that PHI in its original form is a special case of CDHS (Bora et al., 2016).

Cobb–Douglas production function (Cobb and Douglas, 1928) is a 'gold-standard' in production optimization practices (Wu, 1975; Hossain et al., 2012; Hassani, 2012; Saha et al., 2016). Our formulation of CDHS adjusts elasticities via metaheuristic optimization while maintaining global maxima for concavity. The functional form tackles changes in input values (i.e. change in the values of the physical parameters), where maximum CDHS for all exoplanets in the catalog change accordingly, consistent with the database.

[1] Sloan Digital Sky Survey.

[2] Feb. 2019, NASA Exoplanet Archive, https://exoplanetarchive.ipac.caltech.edu.

Constant elasticity earth similarity approach (CEESA). The complexity of the problem involves cross-matching the calculated habitability scores with prediction of class labels. The classification approach does not require explicit computation of the habitability score and, therefore, the efficacy of the overall approach depends on correctness of both approaches; the former being a computational optimization one, and the latter a machine classification one. For example, existing habitability scores/indices do not consider eccentricity as the input variable. To impute the missing values in CDHS requires the introduction of a novel method, since many eccentricity values in the catalog are marked as zero. The approach proposed in this paper – a constant elasticity Earth similarity approach (CEESA) – does that. The imputed values need to be cross-validated with the ones obtained from CEESA, which automatically handles missing values due to the additive form of the model. CEESA is embellished by a novel optimization model, which returns optimal habitability scores for exoplanets, in the process translating the score computation problem to a constrained optimization problem. This is solved by a metaheuristic method, mitigating the complexity and curvature violation issues of gradients and, consequently, producing a method which computes optima. We call this the derivative-free optimization, particularly useful when gradient computation becomes messy (for multi-variate functions such as ours) due to frequent changes in the sign of the function (curvature violation). A cornerstone of the paper is the demonstration of convergence between two approaches, classification approach (prediction of habitability labels) and Earth-similarity approach (CEESA). We applied fuzzy neural networks to solve the habitability class problem. This method is particularly useful when the class membership of objects (exoplanets) is not clear.

The remainder of the paper is organized as follows. We begin by detailing the problem statement and describing several challenges we need to solve in order to arrive at an acceptable solution. In Section 3, we briefly describe the structure of the data present in the PHL-EC catalog and the method used to impute the eccentricity values that are marked '0' in the catalog. We introduce our novel work that estimates the habitability score of exoplanets using CEESA, based on a production function, in Section 4. The section also elaborates on different machine learning (ML) techniques used to implement this model. Section 5 describes a metaheuristic method used to group the planets into different classes. Section 6.4 elaborates a classification technique that has been implemented using fuzzy neural network. The outcome of all methods is shown and justified in Sections 6 and 7 along with their comparison. Related proofs and derivations are included in subsequent sections and appendices.

2. Problem statement

Considering the complexity of assessing the habitability of exoplanets, there is no way to definitively conclude on exoplanets habitability classes, types etc. at this point of time. Hence, it is imperative to explore different methods that can be proved mathematically and whose physical interpretations can be strongly justified. We explore machine learning based classifiers and mathematical models (as metrics) for classification and habitability assessment of newly discovered exoplanets. Our proposed methods try to integrate computational methods, algorithmic learning, and mathematical modeling for determining the degree of habitability of an exoplanet. The outcome of the all the models may be used as indicators while looking for new habitable worlds. Our principal contribution is to propose an integrated approach to habitability classification. The salient features of our work are listed below:

- A new habitability metric is introduced that is capable of accommodating new input parameters with zero or missing values, which was not possible in a product-form formulation, such as CDHS.
- A method was devised capable of handling missing data imputation for eccentricity (restricted to rocky planets).
- The theoretical foundations of the proposed model to compute the Earth Similarity Score are validated.
- The complexity and curvature violation are mitigated by applying a metaheuristic approach.
- Two approaches, CEESA and CDHS, are compared and contrasted.
- We defined neural network and fuzzy neural network-based classification approach, and cross-validated habitability classification outcome with CEESA and CDHS to obtain a more reliable set of potentially habitable exoplanets.

2.1. Justification of the methodology and motivation

In PHL-EC dataset, one of the proposed classification method is sorting all exoplanets into five categories based on their thermal surface characteristics: non-habitable, and potentially habitable: psychroplanet, mesoplanet, thermoplanet and hypopsychroplanet.[3] While it is reasonable in itself to say that many factors of life are dependent on temperature, we believe that while trying to assess the habitability of an exoplanet by means of a metric, more than just the temperature should be taken into consideration. Factors such as radius of a planet, density, escape velocity, eccentricity, and others, are important while determining whether a planet can be potentially habitable or not. For example, there might be cases where the temperature of a planet is in the habitable range, but the planet is too massive to harbor life as we know it, such as e.g. the case of a brown dwarf WD 0806-661B with estimated effective temperature of ∼27 °C (Luhman et al., 2011) (compare to Earth average temperature of ∼15 °C). Hence, developing classification schemes based on only one parameter alone is not sufficient. This has been a prime motivation for the development of metrics such as BCI, PHI, and ESI; this, in turn, inspired us to explore new models that can be used to assess the habitability of exoplanets, which led to the development of CD-HPF. The most significant difference between CD-HPF and the aforementioned habitability metrics is that CD-HPF is inherently adaptive, with the constituent observables in the model having different levels of importance in different planets. Moreover, the overall habitability, as indicated by the CD-HPF, is a score that is maximized on the constituent variables.

The CD-HPF (Bora et al., 2016) is a novel indicator of habitability of an exoplanet. However, the disadvantage of the model is that it is in a multiplicative form. Hence, if the value of an observable is reported as zero, the Cobb–Douglas habitability score (CDHS) of that planet becomes zero, but that is an invalid CDHS for the exoplanet by definition. In this light, it should be noted that none of the observables currently used in the CD-HPF model can have a zero physical value (the values of radius, surface temperature, density, and escape velocity, which are used in the CD-HPF model, cannot be zero for any planet!). To overcome the shortcomings of the CD-HPF, we introduce a new model in this work: the metric in an additive form which can handle naturally occurring, or spurious, zero values of observables. Any number of input parameters can be added to this model. The properties of Constant Elasticity of Substitution (CES) production function motivated us to check the applicability of it in our problem domain. This led to the development of a new metric for habitability,

[3] http://phl.upr.edu/library/notes/athermalplanetaryhabitabilityclassificationforexoplanets.

which we call CEESA: Constant Elasticity of Substitution Earth Similarity Approach. CEESA overcomes the shortcoming of CDHS in handling zero and inconsistent values. While trying to scale up the CD-HPF production function, we faced a bottleneck when we tried to use orbital eccentricity as a feature, because eccentricity of a planet is reported as zero if it is naturally zero or if the data is missing from the database. Eccentricity may play a significant role in determining the habitability of planets. It was proposed earlier that low eccentricity favors multiple planetary systems which, in turn, favors habitability (Limbach and Turner, 2015) as it may control the climate on a planet (Wang et al., 2017). However, most known exoplanets have high eccentricities: two potentially habitable planets, TRAPPIST-1 (Wolf, 2017) and Proxima b, have highly elliptical orbits. It was estimated that, though eccentricity shrinks the HZ, high eccentricity orbit is in certain spin–orbit resonances that can have low effect on planetary climate (Wang et al., 2017). For example, for eccentricity $e = 0.4$ and $p = 0.1$ (where p is the ratio of orbital period to spin period), the HZ is the widest and the climate is most stable. It is obvious that the question is not settled yet, as e.g. our Solar System has a unique feature of very low ellipticities (Earth's orbit is nearly circular at $e = 0.017$), making eccentricity an important parameter.

2.2. List of acronyms and definitions used in the paper

Various acronyms used in the paper are listed below with their full form. For details on each of these terms, please refer to Appendix A. Appendix C contains implementation details on kNN imputation.

- ML: Machine Learning
- PI membership function: (π) membership function.
- Linguistic variable
- Returns to Scale: CRS, DRS and IRS
- CRS: Constant Returns to Scale
- DRS: Decreasing Returns to Scale
- IRS: Increasing Returns to Scale
- CO: Constrained Optimization
- MO: Mathematical Optimization
- kNN: k Nearest Neighbor
- Concavity

3. Data and the catalog

3.1. Classes and features in the dataset

PHL-EC has been created from the Hipparcos catalog which contains 118,219 stars, by examining the information on distances, stellar variability, multiplicity, kinematics, and spectral classification of the listed stars. The reason we use this catalog is because it combines measured and modeled parameters from various sources; it even provides an expanded target list for use by Project Phoenix of the SETI Institute. The PHL-EC dataset constitutes 68 features of about 3875 confirmed exoplanets (at time of writing this paper): 13 categorical features and 55 continuous features. It was discovered that there were 6 features, such as names of the exoplanets, discovery methods and discovery year, which did not play any important role in determining habitability of a planet, and we did not use these.

3.2. Eccentricity data

Further analysis of the dataset revealed that over 60% of the eccentricity values were missing in the catalog (being marked as zero). The eccentricity of a planet is a parameter that determines the amount by which its orbit around another body deviates from a perfect circle. A circular orbit has a value of 0, an elliptical orbit has values between 0 and 1, $e = 1$ is a parabolic orbit, and $e > 1$ defines a hyperbola.

CD-HPF model (Ginde et al., 2016; Saha et al., 2018a), being in a multiplicative form, restricts the values of an observable to be zero as this results in the Cobb–Douglas habitability score (CDHS) of that planet becoming zero, which is invalid by the definition. Hence, the model imposes a constraint that none of the observables currently used in the CD-HPF model can have a zero value (radius, surface temperature, density, escape velocity, or eccentricity). In PHL-EC, missing eccentricity values are assumed as exact 0. Such samples were ignored during previous score calculations by this model to avoid erroneous results. This elimination resulted in disregarding numerous rocky planets possessing Earth-like properties just due to the absence of eccentricity values. Discarding such samples introduces a bias or affects the extent of representation of the results. Ignoring eccentricity as an attribute completely would not work as it may potentially contribute to planet's habitability. In fact, Méndez and Rivera-Valentin (2017) used a mean thermal approximation to show effects of eccentricity on equilibrium temperature and, hence, habitability.

Results, obtained on kNN imputation on few training samples, are tabulated in Table C.15. Some exoplanets which were initially discarded, or found not potentially habitable due to missing or wrong values of eccentricity, were found to be potentially habitable after imputation. Exactly 280 samples with initially missing eccentricity values[4] were found to be potentially habitable on imputation. Thus, imputation of missing eccentricities helped us explore avenues by which the CDH scores can be calculated for a larger number of exoplanets, consequently providing us a with greater number of potentially habitable candidates.

4. CEESA: a new metric for evaluating the habitability of an exoplanet

Having discussed briefly how CDHS could handle missing eccentricity values (an imputation method to be used in CDHS is described in Appendix C), we present our model which handles missing data (eccentricity) inherently. CEESA simply assumes the missing values to be 0. Note that CEESA, being an additive model, does not need to use imputed values while CDHS does![5]

4.1. CES production function

Arrow, Chenery, Minhas and Solow in their now famous paper (Arrow et al., 1961) developed the Constant Elasticity of Substitution (CES) function. This production function with constant elasticity of substitution between the inputs has two major characteristics:

- It is homogeneous of degree one. If we increase the inputs in the CES function by n-fold, output will also increase by n-fold.
- It has a constant elasticity of substitution.

The general form of the Constant Elasticity of Substitution (CES) production function for two inputs is

$$Q(L, K) = (\alpha L^{\rho} + (1 - \alpha)K^{\rho})^{\eta/\rho}, \quad (5)$$

[4] A table containing original and imputed eccentricity values, named 'Original and Imputed values of eccentricity', may be downloaded from http://astrirg.org/projects.html.

[5] For the sake of a fair comparison between CDHS and CEESA, we need to impute missing eccentricity values for use in CDHS and then cross-match CDHS with CEESA values.

where Q = quantity of output, and L, K represent labor and capital, respectively. We define $\rho = \frac{s-1}{s}$; $s = \frac{1}{1-\rho}$, Elasticity of substitution; η = a measure of the economies of scale or elasticity of scale and α = Share parameter.

The Constant Elasticity Earth Similarity Approach (CEESA) is based on the Constant Elasticity of Substitution (CES) production function. Here we considered five parameters to estimate the habitability score of planets, which are: radius, density, surface temperature, escape velocity, and eccentricity. In this production function, the elasticity, ρ, is assumed to be a constant. The CEESA model is shown in Eq. (9). This function is concave if the value of ρ falls in the range: $\rho < 0$, $0 < \rho \leq 1$, and $1 \leq \rho$ and thus a maxima is assured to exist in the range of $0 < \rho \leq 1$. As the values of the constituent parameters across a large sample change over time, the model can adapt to find a value of ρ which will lead the model to find the most potentially habitable planets from a large population.

The motivation for modeling habitability using CES production function is attributed to the following facts: CEESA is additive and, therefore, is resistant to producing zero output if one of the input parameters has zero value. CDHS model by Bora et al. (2016) is limited by this handicap as it produces zero habitability score if any of the input parameters is zero. Additionally, we show that CES production function has CD-HPF, the basis for the CDHS model proposed by Bora et al. (2016), as its limits. Thus, the choice of CES to model habitability score is natural as it is related to the CDHS formulation, with the additional incentive of being endowed with additive form to handle zero eccentricity values. We present the proof of the mathematical relation between CEESA and CDHS in the next subsection.

4.1.1. CEESA yields CDHS in the limiting case

The general form of the Constant Elasticity of Substitution (CES) production function (Hardy et al., 1952; Arrow et al., 1961) for two inputs is

$$Q(L,K) = \gamma\,(\alpha K^{\rho} + (1-\alpha)L^{\rho})^{\eta/\rho}\ ,$$

where Q = quantity of output/CEESA score, and L, K represent input parameters. Define $\rho = \frac{s-1}{s}$; $s = \frac{1}{1-\rho}$, $\rho > 0$. CEESA has as its limits the Cobb–Douglas production function (CDHS), i.e.

$$\lim_{\rho\to\infty} Q = \gamma K^{-\alpha} L^{\alpha-1}\,.$$

Proof. We can rewrite the above equation as

$$Q = \gamma\,(\alpha L^{\rho} + (1-\alpha)K^{\rho})^{\eta/\rho}\ ,$$

$$\frac{1}{\gamma}Q = (\alpha K^{\rho} + (1-\alpha)L^{\rho})^{\eta/\rho} \tag{6}$$

$$\frac{1}{\gamma}Q = \ \exp\left(\eta/\rho \cdot \ln\left[\alpha K^{\rho} + (1-\alpha)L^{\rho}\right]\right) \tag{7}$$

We consider first order Taylor expansion centered at zero of the term inside the logarithm,

$$\alpha K^{\rho} + (1-\alpha)L^{\rho} = \alpha K^{0} + (1-\alpha)L^{0} + \alpha\left(K^{0}\right)^{0} \cdot K^{0} \cdot \ln(K)(\rho-0)$$
$$= (1-\alpha)\left(L^{0}\right)^{0} \cdot L^{0} \cdot \ln(L)(\rho-0) + \frac{(\rho-0)^2}{2!} \cdot f^{2}(x)$$
$$= \alpha + (1-\alpha) + \alpha \cdot \rho \cdot \ln(K) + (1-\alpha) \cdot \rho \cdot \ln(L) + O(\rho^2)$$
$$= 1 + \alpha \cdot \rho \cdot \ln(K) + (1-\alpha) \cdot \rho \cdot \ln(L) + O(\rho^2)\,.$$
$$\alpha K^{\rho} + (1-\alpha)L^{\rho} = 1 + \rho\left[\ln\left(K^{\alpha} \cdot L^{(1-\alpha)}\right)\right] + O(\rho^2)\,. \tag{8}$$

Now, combining Eqs. (6) and (8), we obtain

$$\frac{1}{\gamma}Q = \left[1 + \rho\left(\ln\left(K^{\alpha} \cdot L^{1-\alpha}\right)\right) + O\left(\rho^2\right)\right]^{\eta/\rho}\ .$$

Define $\tau = \frac{1}{\rho}$; $\rho \longrightarrow 0$; $\tau \longrightarrow \infty$. Therefore,

$$\begin{aligned}\lim_{\rho\to 0}\frac{Q}{\gamma} &= \lim_{\tau\to\infty}\frac{Q}{\gamma}\\ &= \lim_{\tau\to\infty}\left(1 + \frac{1}{\tau}\cdot\left[\ln(K^{\alpha}\cdot L^{1-\alpha})\right] + O\left(\tau^{-2}\right)\right)^{\eta\tau}\\ &= \lim_{\tau\to\infty}\left(1 + \frac{1}{\tau}\cdot\left[\ln(K^{\alpha}\cdot L^{1-\alpha})\right]\right)^{\eta\tau}\\ &= \exp\left(\ln(K^{\alpha}\cdot L^{1-\alpha})\right)^{\eta}\ .\end{aligned}$$

Consequently we can write:

$$\lim_{\rho\to 0} Q = \gamma(K^{\alpha} \cdot L^{1-\alpha})^{\eta}$$

Assuming elasticity of scale $\eta = 1$, and constant of elasticity $\gamma = 1$, we get

$$\lim_{\rho\to 0} Q = K^{\alpha} \cdot L^{1-\alpha}\,.$$

This is the CDHS formulation as mentioned in Bora et al. (2016) and is used in this paper with kNN imputation.

4.2. Analytical model

The habitability score, Y is the output of a production function, expressed as a difference between two terms. The first term is nonlinear and if used without any other constraints will yield unbounded habitability scores for all planets. This is the reason we introduced a penalty function which is a linear combination of the same input variables used for the non-linear functional form. The penalty term may be interpreted as a regularizing term to control the growth of the first term, in a constrained optimization framework. The combination of the two terms can be explained in light of Econometric production function and could be interpreted as revenue (non-linear) and cost (linear) respectively. Therefore, the production, Y used to represent habitability score of exoplanets can be thought of as profit. Instead of linearized penalty (cost), we could have non-linear penalty as well. Strictly speaking, this is an analogy with the economic terminologies since the model is derived from Constant Elasticity of Substitution. Without the cost (penalty) term, it would be almost impossible to distinguish exoplanets based on the habitability score since all of them would have unbounded or large values.

CEESA production function for more than two inputs can be written as

$$Y = f\,(R, D, T_s, V_e, E) = \left(r.R^{\rho} + d.D^{\rho} + t.T_s^{\rho} + v.V_e^{\rho} + e.E^{\rho}\right)^{\frac{\eta}{\rho}}\ , \tag{9}$$

where R is radius, D density, T_s surface temperature, V_e escape velocity and E the eccentricity of an exoplanet, which are given (in the dataset); r, d, t, v, and e are the coefficients of radius, density, surface temperature, escape velocity and eccentricity, respectively. The coefficients lie in $(0, 1)$ range, and Y is the target output. The sum of the coefficients r, d, t, v, and e should be 1. The value of η is constrained by the scale of production used: $0 < \eta < 1$ under DRS, and $\eta = 1$ under CRS. Y is the habitability score of exoplanets, where the aim is to maximize Y subject to the constraint that the range of ρ value is $0 < \rho \leq 1$.

Optimization can be conceptualized as a cost against the revenue, which is Y. Here, we consider cost to be a linear combination of the values of the features. Hence, the goal is to minimize cost and to maximize profit. The cost function may be written as the cost for producing Y units i.e.

$$c = w_1 R + w_2 D + w_3 T_s + w_4 V_e + w_5 E\,, \tag{10}$$

where w_1, w_2, w_3, w_4 and w_5 are the weights of the inputs: radius, density, surface temperature, escape velocity and eccentricity, respectively. Thus, the optimization problem becomes

$$\min \{w_1 R + w_2 D + w_3 T_s + w_4 V_e + w_5 E\} \quad \text{subject to } Y\,. \tag{11}$$

The sum of the weights should be 1. The profit function for five parameters is thus

$$\pi = p \cdot Y - w_1 \cdot x_1 - w_2 \cdot x_2 - w_3 \cdot x_3 - w_4 \cdot x_4 - w_5 \cdot x_5\,,$$

where p is the price. We can write the profit function as

$$\pi = pf\,(R, D, T_s, V_e, E) - w_1 R - w_2 D - w_3 T_s - w_4 V_e - w_5 E\,. \tag{12}$$

Profit can be maximized when

$$p\frac{\partial f}{\partial R} = w_1\,, \quad p\frac{\partial f}{\partial D} = w_2\,, \quad p\frac{\partial f}{\partial T_s} = w_3\,,$$

$$p\frac{\partial f}{\partial V_e} = w_4\,, \quad p\frac{\partial f}{\partial E} = w_5\,.$$

The habitability score here is conceptualized as a profit function (Bora et al., 2016).

4.3. Implementation of the model

We applied CES production function to calculate the habitability score of exoplanets. A total of 1644 confirmed rocky exoplanets were taken from the PHL-EC, containing the data for 3689 exoplanets (as of September 2017). Surface temperatures T_s of exoplanets were normalized to the EU (Earth Units) by dividing each of them with Earth's mean surface temperature, 288 K, to avoid zero values in the dataset.

With all input parameters represented in EU, we are looking for the exoplanets whose CEESA score is close to Earth's CEESA score. For each exoplanet, we obtain the optimal elasticity value and the maximum habitability score using gradient-based and metaheuristic methods, noting the limitations of the gradient-based method along the way (see Section 5 for details).

4.3.1. Computation of CES score in DRS and CRS

We have computed elasticity values for CES in the DRS and CRS phases using function *fmincon*, (explained in Appendix E.1). The CES function is applied on varying elasticities to find the CEESA score close to Earth's value (equal to 1). For each exoplanet, we obtain the optimal elasticity and the maximum CEESA value. Table 4 shows a sample of computed values along with the comparison of CEESA score with CDHS. The optimal score for most of the exoplanets for DRS is obtained at $\rho = 0.99$ and for CRS at $\nu = 1.0$ and for CRS at $\rho = \nu = 1.0$.

4.3.2. Meta-heuristic–based optimization

Estimating CEESA scores involves maximizing a production function while observing a set of constraints on the input variables. Under most paradigms, maximizing a continuous function requires calculating a gradient. This may not always be feasible for non-polynomial functions in high-dimensional search spaces. Further, subjecting the input variables to constraints, as needed by CDHS and CEESA, are not always straightforward to represent within the model. Therefore, we implement a novel optimization method to compute the habitability scores of exoplanets (see Section 5 for details).

4.3.3. Classification of exoplanet data using artificial neural network (ANN) and fuzzy logic

Artificial neural network (ANN) is an interconnection of neurons, which are arranged in hierarchical fashion and can be used to solve problems on pattern classification. The architecture of ANN allows weighted interconnections of neurons of input, hidden and output layer, the input pattern is propagated to neurons of hidden layer. Each neuron processes the weighted-input and squashes it to a value between 0 and 1 with the help of the sigmoid activation function. Hidden layer propagates its value to output layer where the neurons again squeeze its value between 0 and 1. The largest value at the output neurons decides the class to which an input pattern belongs. The observed value is compared with the desired value and difference of the two is propagated back to the network, known as learning by Back Propagation. At every iteration, the gradients are computed and weights are updated with the aim of decreasing error. This way, a network is trained for specific outputs and later, the network is used to generalize outputs of test sample. In supervised classification, labels for every element in the universe are known *a-priori*. A fully-connected 3-layered Perceptron architecture is used to classify exoplanets into mesoplanet, psychroplanet and non-habitable classes by considering entries from PHL-EC. Classes with too few samples such as hypopsychroplanet or thermoplanet are excluded since the number of samples is not enough to train accurately for classification purposes.

Classical Sets and Fuzzy sets — A classical set is a collection of distinct elements in which every element posses similar properties. These are sets defined with crisp boundaries. It is defined in such a way that an element is either a member or not a member of the set. For example, a set of days-of-week includes Tuesday, Saturday and, unquestionably, excludes February or December. Accordingly, membership value for an element is 0 or 1 (0 for non-member and 1 for member). Ironically, this is not analogous to the real-world samples where data is uncertain and imprecise. To capture the inexactness of data under study, a concept of fuzzy sets becomes necessary and their usage becomes inevitable. Fuzzy sets, introduced by Zadeh (1965) are an extension of classical sets. By introducing fuzzy logic, we eliminate sharp boundaries, add more details to the data values, thereby facilitating the neural network to learn precisely from the data. Fuzzy sets are sets that evolve around the concept of partial-membership. This implies that an element of fuzzy set may attain a partial membership value between 0 and 1. With the aid of membership function, the crisp feature is converted into multiple fuzzy sets represented in the form of linguistic variables (variables whose values are words in a natural language) as low, medium, high. Essentially, a membership function is a mathematical tool to transform crisp set into fuzzy, and we have used PI (π) membership function to map the crisp features of exoplanets into fuzzy.

The reason behind using fuzzy inputs for classifying exoplanets can be understood by taking a look at a few crucial features, such as radius, eccentricity and density. These features do not give sufficient insight about classes while classifying exoplanets. During the process of classification, they may not be able to help the model to converge quickly with accurate results. Fuzzy representation helps improve classification because of the ability to represent feature values realistically. For exoplanet classification, features are converted into three overlapping fuzzy sets named low, medium and high with the help of PI membership function. The exoplanet dataset comprises 45 features, thus every pattern in the dataset is represented into 45 × 3 vector before being fed into the neural network. Consider an n-dimensional feature vector, $F = [F_1, F_2, F_3 \ldots . F_n]$ consisting of numerical values. Let r be any element in the sample space, λ be the radius of a feature space F_i and c is the central value. Appendix A contains the mathematical definition of the PI (π) function. Every feature F_j of a sample point r can be represented in terms of membership values corresponding to the 3 linguistic variables (low, medium and high). Apparently, the dynamic range of feature space is divided into three overlapping fuzzy sets, each one represented by π functions. λ and c is computed for each of three fuzzy sets, and later membership values are derived for each element r. Assuming F_{jMax} and F_{jMin} are maximum and minimum values of the feature F_j in the sample space, λ and c for the three linguistic spaces can be defined as follows (parameter *fdenom* controls the level of overlap),

$$\lambda_{medium(Fj)} = \frac{1}{2}\left(F_{jMax} - F_{jMin}\right)$$
$$c_{medium(Fj)} = F_{jMin} + \lambda_{medium(Fj)}$$
$$\lambda_{low(Fj)} = \frac{1}{fdenom}\left(c_{medium(Fj)} - F_{jMin}\right)$$
$$c_{low(Fj)} = c_{medium(Fj)} - 0.5\,\lambda_{low(Fj)}$$
$$\lambda_{high(Fj)} = \frac{1}{fdenom}\left(F_{jMax} - c_{medium(Fj)}\right)$$
$$c_{high(Fj)} = c_{medium(Fj)} + 0.5\,\lambda_{high(Fj)}\,.$$

λ and c are computed for each feature and later substituted in π membership function to derive fuzzy values for the PHL-EC dataset. The fuzzy values are then fed into ANN for classification.

The results of classification are cross-matched with CEESA scores of exoplanets. We discussed the convergence of these approaches later in the results section (Section 6). Next section discusses the metaheuristic optimization adopted to compute habitability scores of exoplanets in detail vis-á-vis Particle Swarm Optimization (PSO).

5. Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995) is a biologically inspired metaheuristic for finding the global minima of a function. Traditionally designed for unconstrained inputs, it works by iteratively converging a population of randomly initialized solutions, called particles, toward a globally optimal solution. Each particle in the population keeps track of its current position and the best solution it has encountered, called *pbest*. Each particle also has an associated velocity used to traverse the search space. The swarm keeps track of the overall best solution, called *gbest*. Each iteration of the swarm updates the velocity of the particle toward its *pbest* and the *gbest* values. Let $f(x)$ be the function to be minimized, where x is a d-dimensional vector. $f(x)$ is also called the fitness function. Our focus in this work is restricted to adapting PSO for unconstrained optimization problems to constrained ones as well as mitigating the curvature violation and the complexity of handling multivariate optimization problems. PSO handles this by eliminating the need to compute gradients explicitly.

Curvature violation implies the change of sign in a functional form, i.e. the function changes its shape (from increasing to decreasing, and vice-versa) prematurely even before the optima is reached. Therefore, for the functional forms considered in the habitability model proposed here, the complexity of computing the maximum habitability score involves dealing with 'curvature violations'. The model relies on theoretical guarantees of global optima, and uses the optima to report the maximum habitability score. However, from a practical and computational perspective, we may not obtain the desired optima due to the curvature violation of the functional form. Curvature violation is a major issue in cases of flexible functional form. We expect the global curvature conditions to be consistent with theory when estimations of input parameters and profit function (Y, in this case) are required from a functional form. Along with that, the task of maintaining the flexibility of functional form is also necessary. The phenomenon sometimes arises due to the added local (meaning model-specific, application specific, as opposed to global meaning universalized) restrictions, or constraints, in the optimization problem. Since our habitability score is the solution to a constrained optimization problem, we expect curvature violation due to the general practice of assuming and computing smooth gradients along the functional form. So, if the curve changes sign abruptly, the gradient ascent, which is usually applied to find optima, would fail to detect the violation and report whichever is the highest point of ascent in the curve represented by the function. This complexity is handled by computing global maxima theoretically and algorithmically for each exoplanet, exploiting intrinsic concavity of the functional form, and ensuring 'no curvature violation'. This is explicitly done by the iterative, metaheuristic method (replacing gradient ascent/descent method) described in the next section.

5.1. PSO for constrained optimization

Although PSO (Ricardo, 2008) eliminates the need to estimate the gradient of a function, it still is not suitable for constrained optimization. The standard PSO algorithm does not ensure that the initial solutions are feasible, and neither does it guarantee that the individual solutions will converge to a feasible global solution. Solving the initialization problem is straightforward. We re-sample each random solution from the uniform distribution until every initial solution is feasible. To solve the convergence problem each particle uses another particle's *pbest* value, called *lbest*, instead of its own *pbest* to update its velocity. This is a major deviation from the standard PSO method. Algorithm 1 describes this process.

On each iteration, for each particle, the algorithm first picks two random numbers u_g, u_p. It then selects a *pbest* value from all particles in the swarm that is closest to the position of the particle being updated as its *lbest*. The *lbest* value substitutes $pbest_i$ in the velocity update equation. While updating *pbest* for the particle, the algorithm checks if the current fit is better than *pbest*, and performs the update if the current position satisfies all constraints. The algorithm updates *gbest* as before.

5.2. Representing the problem

In our attempt to discern the habitability scores of discovered exoplanets, we used the PSO algorithm to maximize the objective function. There are several aspects of this approach we look into while considering whether PSO is the right alternative to optimizing a CES production function. To begin with, PSO was not designed to handle constraints in its classical definition. The algorithm needed to be modified to operate in a constrained search space such that the global optima lies within the set of feasible solutions. We also note that, since PSO does not use the gradient of the objective function, it must be able to simulate the gradient in order to gauge whether or not it is generating better solutions at the end of each iteration. Another merit to utilizing PSO for estimating habitability is that we can observe the value of the input variables as it pilots the objective to converge to a globally optimal solution.

A constrained optimization problem can be represented as

$$\underset{x}{\text{minimize}}\, f(x);\ \text{subject to } g_k(x) \le 0,\ \ k = 1\ldots q\,,\ \ h_l(x) = 0,$$
$$l = 1\ldots r\,.$$

```
Algorithm 1: Algorithm for CO by PSO.
Require: $f(x)$, the function to minimize.
Ensure: global minimum of $f(x)$.
for each particle $i \leftarrow 1, n$ do
    repeat
        $p_i \sim U(l,u)^d$
    until $p_i$ satisfies all constraints
    $v_i \sim U(-|u-l|, |u-l|)^d$
    $pbest_i \leftarrow p_i$
end for
$gbest \leftarrow \underset{pbest_i,\, i=1\ldots n}{\text{argmin}}\ f(pbest_i)$
repeat
    $oldbest \leftarrow gbest$
    for each particle $i \leftarrow 1\ldots n$ do
        $u_p, u_g \sim U(0,1)$
        $lbest \leftarrow \underset{pbest_j,\, j=1\ldots n}{\text{argmin}}\ \|pbest_j - p_i\|^2$
        $v_i \leftarrow \omega.v_i + \lambda_g.u_g.(gbest - p_i) + \lambda_p.u_p.(lbest - p_i)$
        $p_i \leftarrow p_i + v_i$
        if $f(p_i) < f(pbest_i)$ and $p_i$ satisfies all constraints then
            $pbest_i \leftarrow p_i$
        end if
    end for
    $gbest \leftarrow \underset{pbest_i,\, i=1\ldots n}{\text{argmin}}\ f(pbest_i)$
until $|oldbest - gbest| < threshold$
return $f(gbest)$
```

Ray and Liew (2001) describe a way to represent non-strict inequality constraints when optimizing using a particle swarm. Strict inequalities and equality constraints need to be converted to non-strict inequalities before being represented in the problem. Introducing an error threshold ϵ converts strict inequalities of the form $g_k{}'(x) < 0$ to non-strict inequalities of the form $g_k(x) = g_k{}'(x) + \epsilon \le 0$. A tolerance τ is used to transform equality constraints to a pair of inequalities,

$$\begin{aligned} g_{(q+l)}(x) &= h_l(x) - \tau \le 0, \qquad l = 1\ldots r\,, \\ g_{(q+r+l)}(x) &= -h_l(x) - \tau \le 0, \qquad l = 1\ldots r\,. \end{aligned}$$

Thus, r equality constraints become $2r$ inequality constraints, raising the total number of constraints to $s = q + 2r$. For each solution p_i, c_i denotes the constraint vector where, $c_{ik} = \max\{g_k(p_i), 0\}$, $k = 1\ldots s$. When $c_{ik} = 0$, $\forall k = 1\ldots s$, the solution p_i lies within the feasible region. When $c_{ik} > 0$, the solution p_i violates the kth constraint.

Under these guidelines, the representation of CDHS estimation under CRS as a CO problem is given below

$$\begin{aligned} &\underset{\alpha,\beta,\gamma,\delta}{\text{minimize}}\text{:}\ Y_i = -R^{\alpha}.D^{\beta}, \\ &\qquad Y_s = -V_e{}^{\gamma}.T_s{}^{\delta} \\ &\text{subject to:} -\phi + \epsilon \le 0, \quad \phi - 1 + \epsilon \le 0\,, \\ &\qquad \forall \phi \in \{\alpha, \beta, \gamma, \delta\} \\ &(\alpha + \beta - 1) - \tau \le 0, \quad (\gamma + \delta - 1) - \tau \le 0\,, \\ &(1 - \alpha - \beta) - \tau \le 0, \quad (1 - \gamma - \delta) - \tau \le 0\,. \end{aligned} \tag{13}$$

Under DRS the last two constraints for Y_i and Y_s are replaced with,

$$\begin{aligned} &\alpha + \beta + \epsilon - 1 \le 0\,, \\ &\gamma + \delta + \epsilon - 1 \le 0\,. \end{aligned} \tag{14}$$

The representation of CEESA score estimation under DRS as a CO problem is given by

$$\begin{aligned} &\underset{r,d,t,v,e,\rho,\eta}{\text{minimize}} \quad Y = -(r.R^{\rho} + d.D^{\rho} + t.T_s{}^{\rho} + v.V_e{}^{\rho} + e.E^{\rho})^{\frac{\eta}{\rho}} \\ &\text{subject to} \qquad -\phi + \epsilon \le 0, \quad \phi - 1 + \epsilon \le 0 \\ &\qquad \forall \phi \in \{r, d, t, v, e, \eta\} \\ &\qquad \rho - 1 \le 0, \quad \rho - 1 + \epsilon \le 0\,, \\ &\qquad (r + d + t + v + e - 1) - \tau \le 0\,, \\ &\qquad (1 - r - d - t - v - e) - \tau \le 0\,. \end{aligned} \tag{15}$$

Under CRS, there is no need for the parameter η (since $\eta = 1$). Thus, the objective function for the problem reduces to,

$$\underset{r,d,t,v,e,\rho}{\text{minimize}} \quad Y = -(r.R^{\rho} + d.D^{\rho} + t.T_s{}^{\rho} + v.V_e{}^{\rho} + e.E^{\rho})^{\frac{1}{\rho}}\,.$$

The CEESA score is thus given by maximizing the objective function,

$$Y = (r.R^{\rho} + d.D^{\rho} + t.T_s{}^{\rho} + v.V_e{}^{\rho} + e.E^{\rho})^{\frac{\eta}{\rho}}\,, \tag{16}$$

where $0 < \rho \le 1$, coefficients r, d, t, v, e lie in $(0, 1)$ and sum up to 1, and η is constrained by the scale of production used: $0 < \eta < 1$ under DRS, and $\eta = 1$ under CRS.

5.3. Handling constraints

As mentioned earlier, the standard PSO algorithm does not guarantee feasible solutions. This is because when particles are initialized or updated, the algorithm does not ensure the resulting solutions are feasible. The solution is twofold, resample each random solution from the uniform distribution until every initial solution is feasible; and while updating velocities always update toward a feasible solution, gathered so far by the algorithm, closest to the particle under update. This ensures that every particle eventually converges toward feasible solutions even if they do not necessarily traverse the feasible solution space.

Incorporating this variation requires the algorithm to store the most optimal feasible solution encountered by each particle in a set, say $L = \{l_1, l_2, \ldots, l_n\}$, as it traverses the search space. At the start of an iteration, for each particle p the algorithm determines the closest position among all solutions in L, called *lbest*, and uses it to update the particle's velocity for the next iteration. Once the iteration is complete, if the particle is within the feasible region, l_p is updated if the new position is a more optimal solution than l_p. Finally, after every particle is updated, the globally best solution is then updated with the best solution in L.

5.4. Simulating the gradient

PSO functions by initializing a set of particles, each with a random position and velocity. The position of a particle describes its solution, which is feasible on initialization. However, the position of the particle is updated on every iteration of the process which might put the particle on an unfeasible solution. At any given time, the algorithm stores a set of locally optimal feasible solutions L and the current globally optimal solution *gbest*. At the start of the process, the algorithm initializes L to the initial positions of the particles and *gbest* to the best solution in L. At each iteration, PSO calculates the distances from the current position of a particle (p) to the current global minima (*gbest*) and to the closest local minima (*lbest*). The algorithm then simulates a gradient based on the sum of these distances and updates the position of the particle. Each iteration can be summed up as

$$v_i = \omega.v_i + k_g(gbest - p_i) + k_p(lbest_i - p_i) \tag{17}$$

$$p_i = p_i + v_i\,, \tag{18}$$

where ω is a constant in $(0, 1)$, and k_g, k_p are uniformly generated random numbers. These values function as inertial weights.

Table 1
CEESA Score Interval under CRS: number of iterations to converge to global optima is reasonably low, implying the function is not stuck in local optima. Moreover, we observe CEESA score in a range. Habitability score should not be a hard number, rather it should lie within a range however small the span may be.

Planet	CEESA score	Score interval		
		Min.	Max.	Delta
Kepler-59 b	178.1611	178.1595	178.1611	0.0016
Kepler-57 c	135.9951	135.9938	135.9951	0.0013
Kepler-61 b	10.0075	10.0065	10.0075	0.0010
Kepler-1393 b	2.24	2.2391	2.2400	0.0009
Kepler-1229 b	1.2604	1.2595	1.2604	0.0009
Kepler-1349 b	1.8369	1.8361	1.8369	0.0009
Kepler-292 d	2.4107	2.4098	2.4107	0.0008
Kepler-901 b	1.555	1.5542	1.5550	0.0008
K2-72 c	1.1933	1.1925	1.1933	0.0008
Kepler-876 b	1.7541	1.7533	1.7541	0.0008

Shi and Eberhart (1998) discussed the use of such weights to regulate velocity, balancing the global and local sections of the simulated gradient. Upper and lower bounds limit the velocity to within $\pm v_{max}$. Once the positions are updated, the algorithm updates L and *gbest* as discussed earlier.

After each iteration, each particle moves a little closer toward *gbest*. This, in turn, leads to L and *gbest* being updated in case any of the particles come across better solutions. Eventually, after several iterations, particles, and their corresponding *lbest* values, converge toward *gbest*. This causes the direction of the simulated gradient to converge toward the actual gradient around the global minima. The corresponding *gbest* is the optimal solution to the problem, in tune with the general principle of PSO exploiting change in position and velocity to converge toward the global optima w.r.t. the parameter ρ of the proposed model. Note that the condition of global minima has been derived theoretically in terms of ρ.

5.5. CEESA score ranges

We noticed that, on average, it requires 89 iterations to obtain global optima under both CRS and DRS constraints. Consider the manner under which the particle swarm converges to the global maxima. After every iteration, *gbest* may be updated toward a more optimal value. However, since the CES production function constructed under CRS or DRS is convex for a given exoplanet the value *gbest* converges toward is always the globally optimal value. Now consider a window into the iterations of the algorithm. Since the objective function is both continuous and convex, the path covered by the best particle of the swarm lies within a continuous interval that, although initially erratic, diminishes as the window moves toward the point of convergence. We observed an average of 89.33 iterations for convergence under CRS, and 89.09 under DRS. We then define a window of 50 iterations, ending at the point of convergence to generate an Earth Similarity Score interval. We list ten planets with the largest intervals in Tables 1 and 2 for CRS and DRS, respectively.[6] Tables illustrate the final converged CEESA scores, and the minimum and maximum values of the defined interval with Delta being the length of the interval.

Table 2
CEESA Score Interval under DRS: number of iterations to converge to global optima is reasonably low implying the function is not stuck in local optima. Moreover, we observe CEESA score in a range. Habitability score should not be a hard number, rather it should lie within a range, however small the span may be.

Planet	CEESA score	Score interval		
		Min.	Max.	Delta
Kepler-163 b	1.8224	1.8198	1.8224	0.0027
Kepler-57 c	191.4616	191.4601	191.4616	0.0015
Kepler-131 c	3.8181	3.8167	3.8181	0.0014
Kepler-409 b	5.6149	5.6137	5.6149	0.0013
Kepler-1263 b	1.5884	1.5874	1.5884	0.00101
Kepler-198 d	2.6528	2.6518	2.6528	0.00101
Kepler-20 c	3.3274	3.3264	3.3274	0.0009
Kepler-171 b	2.0625	2.0616244	2.0625431	0.0009
K2-53 b	2.2593	2.2584	2.2593	0.0008
Kepler-290 b	1.9381	1.9373	1.9381	0.0008

Table 3
Potentially habitable exoplanets considering Earth as reference for CRS ($\nu = 1$, $\rho \leq 1$) and DRS ($\nu < 1$, $\rho \leq 1$):the outcome of CEESA using *fmincon* function.

Exoplanet	Habitability score(CRS)	Habitability score(DRS)
Earth	0.99	0.99
Kepler-186 f	1.15	0.99
Proxima Cen b	1.10	0.99
TRAPPIST-1 e	0.91	0.98
TRAPPIST-1 f	1.02	0.98
Ross 128 b	1.14	1.01

6. Experiment and results

6.1. Result of fmincon function

CEESA scores of a few potentially habitable exoplanets are shown in Table 3 (the full form of the table (4000+ planets) is available as an electronic attachment at http://astrirg.org). The habitability scores are determined for CRS ($\nu = 1$) and DRS ($\nu < 1$) constraints, where the corresponding values of elasticities were found by *fmincon*: $\rho = 1.0$ and $\rho = 0.99$ for CRS and DRS, respectively. We have cross-checked these planets with the database of the potentially habitable worlds — Habitable Exoplanet Catalog (HEC)[7] – and found that these are indeed listed as potentially habitable.

Habitability scores, estimated with CDHS model (without eccentricities) (Bora et al., 2016), are also compared with habitability scores estimated by CEESA model. We have observed that CEESA and CDHS scores are close to each other for exoplanets considered in our experiment. Planets which are considered to be potentially habitable, as per PHL-EC, have habitability score closer to Earth's habitability score computed by these models. Table 4 represents CDHS and CEESA score of some potentially habitable planets calculated using *fmincon* function (without considering the eccentricities).

6.2. Result of PSO

The PSO algorithm is used to estimate CEESA scores for rocky planets. We estimated CEESA scores for both CRS and DRS constraints using the following parameters from the PHL catalog: P. Radius (planet radius), P. Density (planet density), P. Esc Vel (planet escape velocity), P. Ts Mean (planet mean surface temperature) and P. Eccentricity (planet eccentricity). Since surface temperature and eccentricity are not recorded in Earth's units,

[6] The CDHS catalog using imputed values of eccentricity, CEESA Catalogs: CEESA DRS catalog and CEESA CRS Catalog and Original, and imputed values of eccentricity of 1683 rocky exoplanets are available at http://astrirg.org/projects.html.

[7] A derived product from the PHL-EC, also maintained by the PHL; phl.upr.edu/projects/habitable-exoplanets-catalog.

Table 4
Sample simulation Outcome of CDHS and CEESA score for CRS & DRS without considering eccentricity values of planets. Full table is available at http://astrirg.org/projects.html.

Exoplanet	$CEESA_{CRS}$	$CDHS_{CRS}$	$CEESA_{DRS}$	$CDHS_{DRS}$
Earth	0.99	1.00	0.99	1.00
Kepler-20 c	2.40	2.58	2.20	2.31
Kepler-57 c	310.50	314.83	164.00	166.87
Kepler-59 b	263.25	265.21	140.50	142.70
Kepler-61 b	2.01	2.06	1.99	1.91
Kepler-163 b	1.01	1.05	1.00	1.04
Kepler-171 b	2.02	2.26	1.98	2.07
Kepler-186 f	1.15	1.00	0.99	1.00
Kepler-290 b	2.00	2.16	1.90	1.99
Kepler-292 d	2.24	2.14	1.95	1.98
Kepler-1393 b	2.98	3.15	2.95	2.90
Proxima Cen b	1.10	1.09	0.99	1.08
TRAPPIST-1 e	0.91	0.91	0.98	0.97
TRAPPIST-1 f	1.02	0.98	0.98	0.98
Ross 128 b	1.14	1.12	1.01	1.11

Table 5
Sample simulation outcome of CEESA score (CRS & DRS) and CDHS score for (CRS & DRS) with imputed eccentricities. Similarity between imputed CDHS and non-imputed CEESA scores validate the missing value imputation process.

Exoplanet	CRS		DRS	
	$CEESA_{CRS}$	Imputed $CDHS_{CRS}$	$CEESA_{DRS}$	Imputed $CDHS_{DRS}$
Kepler-20 c	2.40	2.55	2.20	2.55
Kepler-59 b	263.25	264.21	140.50	141.70
Kepler-61 b	2.01	2.31	1.99	2.31
Kepler-163 b	1.01	1.89	1.00	1.89
Kepler-171 b	2.02	2.89	1.98	2.89
Kepler-186 f	1.15	1.15	0.99	1.15
Kepler-290 b	2.00	2.44	1.90	2.44
Kepler-292 d	2.24	2.54	1.95	2.54
Kepler-1393 b	2.98	2.32	2.95	2.43
Proxima Cen b	1.10	1.09	0.99	1.09
TRAPPIST-1 e	0.91	0.91	0.98	0.99
TRAPPIST-1 f	1.02	0.92	0.98	1.02

we normalized these values by dividing them with Earth's surface temperature (288 K) and eccentricity (0.017), respectively. PHL-EC records empty values for planets whose surface temperature is not known. We chose to drop such records from our experiment. Catalog also assumes zero eccentricity for those planets where the data is not available. For such planets we employ data imputation (see next subsection).

Our experiment used $n = 25$ particles to traverse the search space, with learning rates $\lambda_g = 0.8$ and $\lambda_p = 0.2$. It used an integral weight of $\omega = 0.6$ and upper and lower bounds ± 1.0. We used an error threshold of $\epsilon = 1 \times 10^{-6}$ to convert strict inequalities to non-strict inequalities, and a tolerance of $\tau = 1 \times 10^{-7}$ to transform an equality constraint to a pair of inequalities. Tables C.17a and C.17b in Appendix C show CEESA scores for a sample of exoplanets obtained under CRS and DRS constraints, respectively.

6.3. Comparison of CEESA score with imputed CDHS

Eccentricity values are not available in the PHL-EC for all exoplanets. Unknown eccentricity values are set to 0. We did not have to impute eccentricity values to compute CEESA scores. As explained earlier, only CDHS needs imputed eccentricity values (see Appendix C for details on eccentricity imputation). Table 5 shows calculated CEESA and CDHS scores for several exoplanets. Since CES model is an additive model, we have used catalog's eccentricity values of planets to estimate the CEESA score. However, the validity of the imputation method is easily verified from Table 5, as we observe small difference between CEESA (no eccentricity imputation) and CDHS (with imputed eccentricity). This is also evident from the Root-Mean-Square Error (RMSE) plot in Fig. C.3 (App. C). Table 6 shows the correspondence between habitability scores computed using CDHS and CEESA models along with their predicted class labels (using neural nets and fuzzy neural nets).

6.4. Experiments with fuzzy and non-fuzzy ANN

On various combinations of feature sets, the network was trained and tested for two main settings. In the first setting the crisp inputs are applied, and in the second, fuzzy inputs are fed to the network. To fuzzify the sample space, the values of λ_{low}, λ_{medium}, λ_{high}, C_{low}, C_{medium} and C_{high} are calculated for every feature in the feature set. Feature values of each sample are then converted into fuzzy component by using PI membership mentioned in Section 4.3.3 (see Appendix A for details), classification of exoplanet data using Artificial Neural Network (ANN), and Fuzzy Logic. Non-habitable planets, mesoplanets and psychroplanets are labeled as Class 1, Class 2 and Class 3, respectively.[8] A case-by-case explanation (cases 1–8) of every feature set is explored (Tables 7–14 show class-wise performance measure). The different measures used are Accuracy, Precision, Recall, Sensitivity, Specificity and F-score (Powers, 2011). The classification problem had several challenges including imbalanced data. Each of these measures has its follies. This is the reason we reported all possible performance measures to establish the efficacy of our method. For example, Fscore of 1 implies perfect classification despite the presence of class imbalance. All of these measures range from 0 to 1 depending on the performance from bad to excellent.

Case 1 (3-class dataset): The dataset comprises 45 features to classify exoplanets into 3 classes, namely non-habitable planets (Class 1), mesoplanets (Class 2), and psychroplanets (Class 3). The network consists of fully connected layers of neurons comprising 45 input, 20 hidden and 3 output neurons. Weights of the interconnections are randomly initialized and back propagation algorithm trains the network to update the weights thus minimizing the mean square error. Learning rate is tuned to 0.015 and number of epochs is set to 500. The classification results are shown in Table 7.

Case 2 (3-class dataset): The dataset is same as above. The only difference is the inclusion of preprocessing step that converts the n-dimensional feature into $3n$-dimensional linguistic pattern by using PI membership value. After conversion of data into fuzzy feature space, the network is fed with 135 inputs, which gets propagated to 20 neurons in hidden and 3 neurons in output layer. The weights are initialized with small random values, and the other parameters are kept same. The class-wise classification results are shown below. Apparently, the results are better than the one without fuzzy inputs (Case 1). Table 8 shows the result for Case 2, and Fig. 1 shows the ROC curves for Class 3 and Class 2 samples.

Case 3 (2-class dataset): This particular case contains 'Two-class dataset' (mesoplanets and psychroplanets combined into class 2 and non-habitable planets as class 1). Data is without fuzzy inputs and all parameters and features are same as in Case 1. Table 9 shows the result of Case 3.

Case 4 (2-class dataset): We consider two classes again (identical to Case 3) with fuzzy inputs. The parameters are same as Case 2. The result is shown in Table 10.

Case 5 (3-class dataset): This case considers the same 3 classes used in Cases 1–2. A new combination of features, consisting of

[8] 3-class dataset: dataset containing class labels 1, 2 and 3 representing non-habitable planets, mesoplanes and psychroplanets respectively; 2-class data set: dataset containing class labels 1 and 2 representing non-habitable planets and potentially habitable planets (mesoplanets and psychroplanets collapsed in to class 2) respectively.

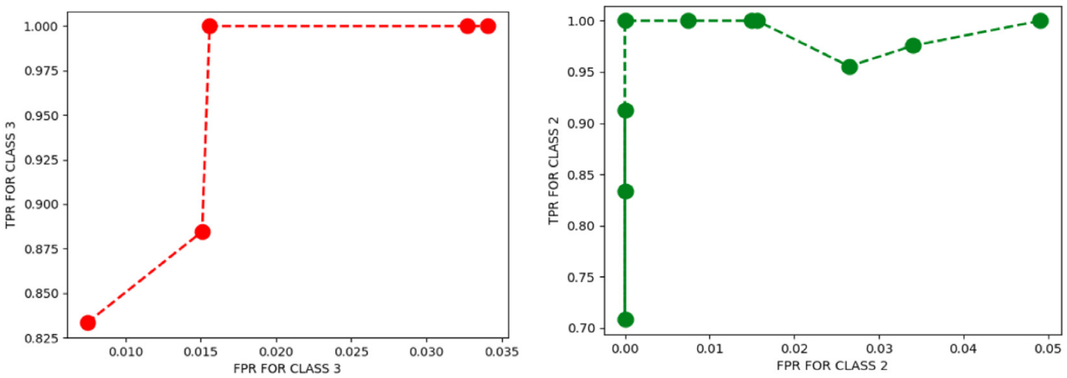


Fig. 1. ROC plot for class 3 samples (*Left*), and ROC for class 2 samples (*Right*) of fuzzy classification. A sample of ROC plots is shown class-wise.

Table 6
Summary of results of both methods: matching of CEESA scores with predicted classes. For example, TRAPPIST-1 e, labeled as psychroplanet by fuzzy neural net (Method 2) with 100% accuracy, also has both CDHS and CEESA scores close to Earth (i.e. 1).

Exoplanet	Method 1: explicit score calculation				Method 2: NN/fuzzy-NN classification	
	$CDHS_{DRS}$	$CDHS_{CRS}$	$CEESA_{DRS}$	$CEESA_{CRS}$	Accuracy (%)	Predicted class
Proxima Cen b	1.08	1.09	0.99	1.10	100.00	psychroplanet
TRAPPIST-1 c	1.14	1.16	1.06	1.19	96.40	non-habitable
TRAPPIST-1 d	0.96	0.89	0.98	0.99	100.00	mesoplanet
TRAPPIST-1 e	0.97	0.91	0.98	0.91	100.00	psychroplanet
TRAPPIST-1 f	0.98	0.98	0.98	1.02	99.70	psychroplanet
TRAPPIST-1 g	1.09	1.11	0.99	1.11	92.30	psychroplanet

Table 7
Case 1: Result of fuzzy classification for 3-class dataset without fuzzy inputs.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	1.000	1.000	1.000	1.000	1.000	1.000
2	0.976	0.943	0.895	0.895	0.989	0.917
3	0.976	0.933	0.966	0.966	0.981	0.948

Table 8
Case 2: Result of fuzzy classification of 3-class dataset with fuzzy inputs.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	1.000	1.000	1.000	1.000	1.000	1.000
2	0.995	1.000	0.967	0.967	1.000	0.982
3	0.995	0.980	1.000	1.000	0.993	0.989

Table 9
Case 3: Result of fuzzy classification for 2-class dataset without fuzzy inputs.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	1	1	1	1	1	1
2	1	1	1	1	1	1

Table 10
Case 4: Result of fuzzy classification for 2-class dataset with fuzzy inputs.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	1	1	1	1	1	1
2	1	1	1	1	1	1

mass, radius, minimum mass and composition class values, is supplied to neural network of 4 input, 4 hidden and 3 output neurons. Though the network was able to perform classification at a decent level, the accuracy, precision and recall values were not as good as what was obtained with all-feature dataset (Case 1). Looking at the accuracy, one can infer that, even if the rest of the features are not used, planet's mass and radius are adequately good features that can separate the three classes. Learning rate was tuned to 0.2, and obtained classification accuracy is shown in Table 11.

Case 6 (3-class dataset): The same dataset (as Case 5) with 12 fuzzy inputs is run on network of 6 hidden neurons and the learning rate is tuned to 0.19. With these set of parameters, class-wise results shows that the network is unable to classify exoplanets at a satisfactory level. Case 5 suggests that mass and radius are good enough features to classify exoplanets, but their fuzzy values could not add enough information for the network to behave as an exemplary classifier. Table 12 shows the classification accuracy of this case.

Case 7 (3-class dataset): A different set of features, from which planet's surface temperature is removed, is applied to the network. The network now consists of 42 input, 11 hidden and 3 output neurons. The classification results are very encouraging. The insight with regard to habitability is, although feature like surface temperature can clearly demarcate exoplanets, but parameters like mass, radius, eccentricities, when blended together, can also bring impeccable accuracy during classification. The accuracy after 400 epochs is shown in Table 13.

Case 8 (3-class dataset): A fuzzy set built from the dataset of Case 7 is used for classification for a network with 19 hidden

Table 11
Case 5: Result of fuzzy classification for 3-class dataset with reduced set of parameters.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	0.888	0.975	0.808	0.808	0.978	0.978
2	0.825	0.534	0.398	0.398	0.925	0.439
3	0.751	0.814	0.814	0.814	0.724	0.653

Table 12
Case 6: Result of fuzzy classification for 3-class dataset with reduced parameters and fuzzified inputs.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	0.480	0.488	0.887	0.887	0.092	0.625
2	0.816	nan	0.058	0.058	0.998	nan
3	0.659	nan	0.109	0.109	0.917	nan

Table 13
Case 7: Result of fuzzy classification for 3-class dataset. Surface temperature is not considered as input parameter.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	0.999	0.994	1.000	1.000	0.998	0.997
2	0.999	0.995	1.000	1.000	0.998	0.998
3	0.997	1.000	0.994	0.994	1.000	0.997

Table 14
Case 8: Result of fuzzy classification for 3-class dataset.

Class	Accuracy	Precision	Recall	Sensitivity	Specificity	Fscore
1	0.997	1.000	0.993	0.993	1.000	0.997
2	0.999	0.995	1.000	1.000	0.998	0.998
3	0.999	0.996	1.000	1.000	0.998	0.998

neurons (other parameters are kept same). The model has shown decent accuracy (reflected in Table 14).

7. Discussion and conclusion

The concept of developing a classifier based on our growing knowledge of exoplanets is fascinating as it draws inferences from two different approaches, Earth similarity and habitability. We provide a manuscript that develops a tool for planetary habitability using known functions to score habitability combined with planetary features to generate a predictor. The predictor is developed as a computational intelligence (CI) and classification approach. Both approaches produce a similar outcome.

PSO is used to track dynamic functions of the type that allow for the oscillation that we also mitigate. Additionally, we use different set of features (full and restricted) to test the efficacy of our classifiers (Cases 1–8, Tables 7–14). The results suggest that the use of Proxima b and TRAPPIST-1 for training, and remaining samples in the catalog for testing, performed well.

We introduce CEESA, a novel model based metric that defines the habitability score of exoplanets. The strength of this kind of modeling is that it can naturally handle missing data or data points with zero values. The motivation behind attempting to develop metrics for habitability in this manner is to be able to observe trends from incomplete or unavailable data to the best of technological ability, and CEESA model can naturally accomplish that. The model is scalable and can be extended to accommodate more planetary observables (the proof is included in Appendix B). Additionally, if ρ in Eq. (5) approaches zero in the limit, we obtain the Cobb–Douglas production function. However, computing the optima of such a multi-variate model posed significant computational challenges (curvature violation and oscillating local minima). The practical consequence of 'curvature violation and oscillating local minima' is the premature convergence of the habitability scores of exoplanets. Since most approaches to optimization are gradient-based, oscillating local minima sometimes give the false impression that we have reached an optima when, in fact, we have not. This can be thought of as 'local optima groove', where the gradient of the functional form is stuck and, therefore, is forced to converge prematurely even when the global optima exists otherwise. The direct impact of this on the habitability computation is the production of sub-optimal habitability score.

The habitability score problem is therefore interpreted as a constrained optimization problem, solved by Particle Swarm Optimization. Especially noticeable about Particle Swarm Optimization is the lack of the need for a gradient, allowing PSO to work in high-dimensional search spaces with a large number of constraints to estimate precise habitable score. Further, particles of the swarm in most implementations operate independently during each iteration, their updates can occur simultaneously and even asynchronously, yielding much faster execution times than descent/ascent type methods. Using PSO to calculate the habitability scores is beneficial when the number of input parameters is large, which further increases the number of constraints, resulting in a model too unfeasible for traditional optimization methods.

We have used ML methods (Saha et al., 2018a,b,c) and mathematical modeling to develop richer inference from the data of exoplanets, which can bolster our understanding of factors that affect habitability in the long run. The classification method used here draws the advantages of both neural network and fuzzy logic. We considered mesoplanets, psychroplanets and non-habitable planets as the class labels in the dataset. PI membership function helps the algorithm to assign membership to each data point. It was observed that the classifier worked well when all the parameters are used for the classification, rather than using a few. The accuracy of the classifier is above 95% both with and without fuzzy inputs. Additionally, the fuzzy approach on this problem is an insightful attempt, as planets may have membership in all class labels but just to differing degrees. Given the sparsity of our knowledge about planets, their features, and habitability, a fuzzy approach seemed more suitable than more traditional classification approaches.

Our proposed model CEESA used eccentricity as the fifth parameter to compute habitability scores of planets. Use of eccentricity is not in practice in any of the previous indices, including ESI and PHI estimation of planets. Even a previous model, CDHS (Bora et al., 2016), used the same parameters as in ESI and PHI to compute habitability scores. However, even though CDHS model is scalable and can, in principle, handle any number of parameters, handling zero eccentricity values rendered the metric unfeasible to use because of the product nature of the model. This is not the case in CEESA. We have cross-checked CEESA scores with imputed CDHS model (where kNN imputation was used to fill in for zero eccentricity values in the product formulation of CDHS). Tables 3–6 show the outcome of both models. Imputation of missing values in CDHS and natural mitigation of those values in CEESA are two major contributions of the current work. This computational approach is further bolstered by computing optimal habitability scores without having to compute the gradient explicitly, an important step towards derivative-free optimization.

Acknowledgments

This research has made use of the PHL's Exoplanet Catalog, maintained by the Planetary Habitability Laboratory at the University of Puerto Rico at Arecibo, and NASA Astrophysics Data System Abstract Service. We would like to thank the Science and Engineering research Board (SERB), Department of Science and Technology, Government of India, for supporting our research by providing us with resources to conduct our experiments. The project reference number is: EMR/2016/005687.

Appendix A. Definition of key terms used in CEESA model

This section defines few key terms used in the paper.

- **Mathematical Optimization** Optimization is one of the procedures to select the best element from a set of available alternatives in the field of mathematics, computer science, economics, or management science (Hájková and Hurnik, 2007). An optimization problem can be represented in various ways. Below is the representation of an optimization problem. Given a function $f : A \to R$ from a set A to the real numbers R. If an element x_0 in A is such that $f(x_0) \leq f(x)$ for all x in A, this ensures minimization. The case $f(x_0) \geq f(x)$ for all x in A is the specific case of maximization. The optimization technique is particularly useful for modeling the habitability score in our case. In the above formulation, the domain A is called a search space of the function f, CD-HPF in our case, and elements of A are called the candidate solutions, or feasible solutions. The function as defined by us is a utility function, yielding the habitability score CDHS. It is a feasible solution that maximizes the objective function, and is called an optimal solution under the constraints known as **Returns to scale**.
- **Returns to scale** measure the extent of an additional output obtained when all input factors change proportionally. There are three types of returns to scale:
  1. **Increasing returns to scale (IRS)**. In this case, the output increases by a larger proportion than the increase in inputs during the production process. For example, when we multiply the amount of every input by the number N, the factor by which output increases is more than N. This change occurs as
     - (i) Greater application of the variable factor ensures better utilization of the fixed factor.
     - (ii) Better division of the variable factor.
     - (iii) It improves coordination between the factors.
  2. **Decreasing returns to scale (DRS)**. Here, the proportion of increase in input increases the output, but in lower ratio, during the production process. For example, when we multiply the amount of every input by the number N, the factor by which output increases is less than N. This happens because:
     - (i) As more and more units of a variable factor are combined with the fixed factor, the latter gets over-utilized. Hence, the rate of corresponding growth of output goes on diminishing.
     - (ii) Factors of production are imperfect substitutes of each other. The divisibility of their units is not comparable.
     - (iii) The coordination between factors get distorted so that marginal product of the variable factor declines.
  3. **Constant returns to scale (CRS)**. Here, the proportion of increase in input increases output in the same ratio, during the production process. For example, when we multiply the amount of every input by a number N, the resulting output is multiplied by N. This phase happens for a negligible period of time and can be considered as a passing phase between IRS and DRS.
- **Computational Techniques in Optimization (CO)**. These are a broad family of approximation techniques used to compute values of functions, optima of functions, root finding problems, fixed point iterations etc. The computational optimization (CO) technique described in the paper is PSO where the focus was to replace gradient computations with gradient emulation. There exist several well-known techniques including Simplex, Newton-like and Interior point-based techniques (Nemirovski and Todd, 2008). One such technique is implemented via MATLAB's optimization toolbox using the function *fmincon*. This function helps find the global optima of a constrained optimization problem which is relevant to the model proposed and implemented by the authors. Illustration of the function and its syntax are provided in Appendix E.1. It is important to note that MATLAB deploys a suite of optimization techniques in its library such as active set, interior point, metaheuristics and evolutionary techniques to handle a variety of functions namely convex/concave, non-concave non-convex, smooth and non-smooth etc.
- **Concavity**. Concavity ensures global maxima, theoretically. The implication of this fact in our problem statement and solution approach is that if CD-HPF is proved to be concave under some constraints (elaborated in the paper, Section 5 and Appendix B), we are guaranteed to have maximum habitability score for each exoplanet in the global search space. This is particularly useful for the metaheuristic optimization (approximation of global optima in Section 5) approach adopted in the paper as it is easy to verify the veracity of the proposed approach against known optimal solutions guaranteed by concavity.
- ***k*-Nearest Neighbor (kNN)**: kNN is a classification algorithm, that works as follows. Given a parameter k, find the k nearest neighbors, and take a majority vote from their classes. kNN performs reasonably well with binary classification, and typical values of k are around 5. kNN may be used as regression technique where the weights of

nearby points are used to predict a neighboring point. This is the same principle we used to impute missing values. kNN based imputation (regression) is powerful as it handles non-linearity quite well and does not need assumptions of normality in error terms.

- **Machine Learning**. Classification of patterns based on data is a prominent and critical component of machine learning and will be highlighted in subsequent part of our work where we made use of a standard kNN algorithm. The algorithm is modified to tailor to the complexity and efficacy of the proposed solution. Optimization, as mentioned above, is the art of finding maximum and minimum of surfaces that arise in models utilized in science and engineering. More often than not, the optimum has to be found in an efficient manner, i.e. both the speed of convergence and the order of accuracy should be appreciably good. Machines are trained to do this job as, most of the times, the learning process is iterative. Machine learning is a set of methods and techniques that are intertwined with optimization techniques. The learning rate could be accelerated as well, making optimization problems deeply relevant and complementary to machine learning.
- **Pi membership function**. A membership function is an arbitrary curve that maps every value in the input space between 0 and 1. If X is the universe of discourse, x denotes an element, $\mu_A(x)$ is the membership function of x in A, then membership value is represented as $A = (\mu_A(x), x)$. The PI (π) function for a sample r (with c and λ as centre and radius of the dataset), can be defined as:

$$\pi(r; c, \lambda) = \begin{cases} 2(1 - \frac{\|r-c\|}{\lambda})^2 & \text{for} \quad \lambda/2 \leq \|r-c\| \leq \lambda \\ 1 - 2(\frac{\|r-c\|}{\lambda})^2 & \text{for} \quad 0 \leq \|r-c\| \leq \lambda/2 \\ 0 & \quad \text{Otherwise} \end{cases}$$

Fig. A.2 illustrates formation of 3 overlapping fuzzy sets using PI membership function.

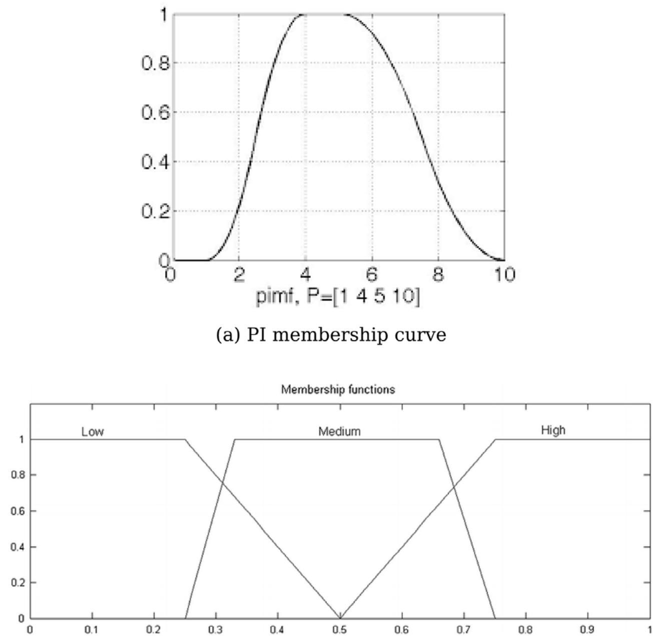
Appendix B. The proof of CES model scalability

Here we prove optimality using Hessian matrices. If a Hessian matrix of a function is symmetric about its primary diagonal, a global optimum exists for that function. The general form of a Hessian matrix for a function is given by

$$\text{Hess}\,(Y) = \begin{bmatrix} \frac{\partial^2 Y}{\partial A^2} & \frac{\partial^2 Y}{\partial B \partial A} \\ \frac{\partial^2 Y}{\partial A \partial B} & \frac{\partial^2 Y}{\partial B^2} \end{bmatrix}. \tag{19}$$

Here, the elements of Hess (Y) are given as

$$\begin{aligned} \frac{\partial^2 Y}{\partial A^2} &= ka\eta\Big[(\rho-1)A^{\rho-2}(aA^\rho + bB^\rho)^{\frac{\eta-\rho}{\rho}} \\ &\quad + \frac{\eta-\rho}{\rho}A^{\rho-1}\,(aA^\rho + bB^\rho)^{\frac{\eta-2\rho}{\rho}}\Big], \\ \frac{\partial^2 Y}{\partial B \partial A} &= kab\eta\,(\eta-\rho)\,A^{\rho-1}B^{\rho-1}\,(aA^\rho + bB^\rho)^{\frac{\eta-2\rho}{\rho}}, \\ \frac{\partial^2 Y}{\partial A \partial B} &= kab\eta\,(\eta-\rho)\,A^{\rho-1}B^{\rho-1}\,(aA^\rho + bB^\rho)^{\frac{\eta-2\rho}{\rho}}, \\ \frac{\partial^2 Y}{\partial B^2} &= ka\eta\Big[(\rho-1)B^{\rho-2}\,(aA^\rho + bB^\rho)^{\frac{\eta-\rho}{\rho}} \\ &\quad + \frac{\eta-\rho}{\rho}B^{\rho-1}\,(aA^\rho + bB^\rho)^{\frac{\eta-2\rho}{\rho}}\Big]. \end{aligned} \tag{20}$$



(a) PI membership curve

(b) PI membership with three overlapping sets

Fig. A.2. PI membership Curve.

From Eqs. (20), we can see that

$$\frac{\partial^2 Y}{\partial B \partial A} = \frac{\partial^2 Y}{\partial A \partial B}.$$

This implies that Hess (Y) is symmetric about the primary diagonal, and hence, Y has a global optimum. For a CES production function with n terms, the general form of the elements of Hess (Y) is given as

If $i = j$,

$$\begin{aligned} a_{ij} = k\eta\alpha_i \Bigg[&(\rho-1)A_i^{\rho-2}\left(\sum_{m=1}^{n}\alpha_m A_m^\rho\right)^{\frac{\eta-\rho}{\rho}} \\ &+ \frac{(\eta-\rho)}{\rho}A_i^{\rho-1}\left(\sum_{m=1}^{n}\alpha_m A_m^\rho\right)^{\frac{\eta-2\rho}{\rho}}\Bigg]. \end{aligned} \tag{21}$$

If $i \neq j$,

$$a_{ij} = k\eta(\eta-\rho)\alpha_i\alpha_j A_i^{\rho-1}A_j^{\rho-1}\left(\sum_{m=1}^{n}\alpha_m A_m^\rho\right)^{\frac{\eta-2\rho}{\rho}}. \tag{22}$$

$\forall\, 1 \leq i \leq n, 1 \leq j \leq n$; α_i is the ith coefficient and A_i is the ith parameter. For any $n \in \{1, 2, 3, \ldots\}$, the element in the $(i, j)^{th}$ position of the Hessian matrix is given by Eqs. (21) and (22). From Eq. (22), it is evident that all of the non-diagonal elements are symmetric about (i, j). Hence, the Hessian matrix of a CES production function with any number of variables is always symmetric about the primary diagonal.

Thus, we conclude that the CES function has a global optimum, and is scalable for any $n \in \{1, 2, 3, \ldots\}$.

B.1. Constraint conditions for elasticities: ESI with dynamic input elasticity fails to be an optimizer

The function, where R_e, D_e, T_e and V_e are the radius, density, surface temperature and escape velocity of Earth and are constant terms, is written as

$$Y = k\left(1 - \frac{R_e - R}{R_e + R}\right)^{\alpha}\left(1 - \frac{D_e - D}{D_e + D}\right)^{\beta} \times \left(1 - \frac{T_e - T}{T_e + T}\right)^{\gamma}\left(1 - \frac{V_e - V}{V_e + V}\right)^{\delta} . \tag{23}$$

Here, R, D, T and V are radius, density, surface temperature and escape velocity, respectively, of the planets under study, and k is a constant parameter. Differentiating Eq. (23) partially with respect to R,

$$\frac{\partial Y}{\partial R} = \alpha\left(\frac{2R}{R_e + R}\right)^{\alpha - 1} \frac{2R_e}{\left(R_e + R\right)^2} , \tag{24}$$

and finding the second partial derivative of Eq. (24), we obtain

$$\begin{aligned} \frac{\partial^2 Y}{\partial R^2} &= \alpha(\alpha - 1)\left(\frac{2R}{R_e + R}\right)^{\alpha - 2}\left(\frac{2R_e}{(R_e + R)^2}\right)\left(\frac{2R_e}{(R_e + R)^2}\right) \\ &\quad - \alpha\left(\frac{2R}{R_e + R}\right)^{\alpha - 1}\left(\frac{4R_e(R_e + R)}{(R_e + R)^4}\right) \\ &= \alpha(\alpha - 1)\left(\frac{2R}{R_e + R}\right)^{\alpha - 2}\frac{4R_e^2}{(R_e + R)^4} - \alpha\left(\frac{2R}{R_e + R}\right)^{\alpha - 1} \\ &\quad \times \left(\frac{4R_e^2 + 4R_e R}{(R_e + R)^4}\right) \\ &= \alpha(\alpha - 1)\left(\frac{2R}{R_e + R}\right)^{\alpha - 2}\frac{1}{(R_e + R)^4} \\ &\quad \times \left((\alpha - 1)4R_e^2 - \frac{2R}{R_e + R}(4R_e^2 + 4RR_e)\right) \\ &= \alpha(\alpha - 1)\left(\frac{2R}{R_e + R}\right)^{\alpha - 2}\frac{1}{(R_e + R)^4}\left((\alpha - 1)4R_e^2 - 8RR_e\right) . \end{aligned} \tag{25}$$

For concavity, the second partial derivative must be greater than zero. Thus, relating this to Eq. (25), we obtain

$$\alpha\,(\alpha - 1)\left(\frac{2R}{R_e + R}\right)^{\alpha - 2}\frac{1}{(R_e + R)^4}\left((\alpha - 1)4R_e^2 - 8RR_e\right) > 0 . \tag{26}$$

Upon simplifying the inequality above, we arrive at

$$\alpha - 1 > 2\frac{R}{R_e} . \tag{27}$$

Generalizing the result in Eq. (27) for all variables in the data and the corresponding elasticities, we arrive at the following results,

$$\begin{aligned} \beta - 1 &> 2\frac{D}{D_e} , \\ \gamma - 1 &> 2\frac{T}{T_e} , \\ \delta - 1 &> 2\frac{V}{V_e} . \end{aligned} \tag{28}$$

Summing up the results presented in Eqs. (27) and (28), we finally derive the following relationship,

$$\alpha + \beta + \gamma + \delta > 2\left(\frac{R}{R_e} + \frac{D}{D_e} + \frac{T}{T_e} + \frac{V}{V_e}\right) + 4 . \tag{29}$$

Eq. (29) shows that the sum of the four elasticity constants cannot be less than or equal to 1 (in fact, cannot be less than 1). This is the case of IRS (increasing return to scale) in CDHPF function, which means that function is neither concave nor convex. The new metric holds for IRS condition only, which does not ensure a global maxima, implying lack of theoretical foundation for the ESI input structure.

Appendix C. Imputation of missing values of eccentricity

One of the classification schemes proposed by the PHL was selection of planets into habitability classes based on their surface temperatures.[9] However despite surface temperature being a crucial parameter to compute habitability metric, factors such as the radius of a planet, density, escape velocity, eccentricity, etc. are also important. Developing classification schemes based on only one parameter alone is not sufficient as this would involved just comparisons. This has been a prime motivation for the development of metrics such as BCI, PHI, and ESI; this, in turn, inspired us to explore models that can be used to assess the habitability of exoplanets, which led to the development of CD-HPF.

Orbital eccentricity of an astronomical object is a parameter that determines the amount by which its orbit around another body deviates from a perfect circle. In this section, we present our methods to compute the missing values of eccentricity of rocky planets in the given exoplanet catalog. We ignored gaseous planets from our consideration because of the improbability of them being habitable. The catalog contains 1696 rocky planets out of which 1537 planet's eccentricities marked as zero, presumably missing values. However, eccentricity of a planet affects the climate, atmosphere, and the composition of a planet to a large degree, and can be an important factor in habitability (e.g. Wang et al., 2017).

C.1. Preprocessing: Dimensionality reduction

Incremental principal component analysis was used to reduce the number of features. This algorithm assigns weights to every feature. Features like luminosity or number of moons do not play as large a role in computing the eccentricity of a planet as does the mass and the number of neighboring planets. The primary features contributing to eccentricity of the planet include:

- Zone Class
- Mass Class
- Atmosphere Class
- Composition Class
- Mass of the planet
- Density

We convert these categorical attributes to numeric, remove tuples which do not contain these determining parameters, and form a subset to be used for imputing.

C.2. Normalization

Surface temperature and eccentricity are the only two attributes not expressed in Earth Units. kNN computes similarity scores by giving equal weightage to all attributes, so we scale these two attributes by dividing each of them by the Earth's value to avoid inconsistent results. We also scale down the eccentricity

[9] http://phl.upr.edu/library/notes/athermalplanetaryhabitabilityclassificationforexoplanets.

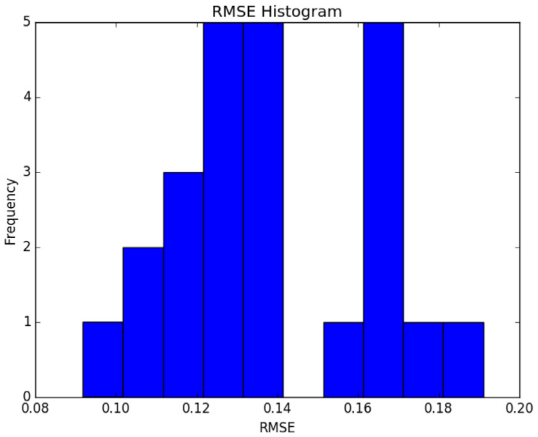


Fig. C.3. RMSE obtained on different iterations for different folds of the dataset.

after imputation. This is essential to keep the habitability metric close to 1.

C.3. kNN-based imputation

Imputation is a technique to avoid pitfalls involved with tuple deletion of cases that contain one or more missing values. It retains all cases by replacing missing data with an estimated value based on other available parameters which influence it. There are several estimation techniques to select from on the basis of relationship between the parameters. These include imputation based on mean, median and mode or by regression.

In kNN-based imputation method, the k nearest neighbors of the object with missing values are used to impute the missing values in the object. The neighbors are determined based on a similarity metric. It chooses neighbors by assigning weights to samples using the mean squared difference on features for which two rows both have observed data. The assumption behind using kNN for missing values is that a point value can be approximated by the values of the points that are closest to it, based on other variables. It works well due to the strong relationship between the known attribute values and missing values in a sample as all of these values contribute to the distance metric. kNN works best on a low dimensional dataset. Thus, kNN algorithm for imputing is applied to the preprocessed dataset, obtained on performing dimensionality reduction using PCA.

Data with known values of eccentricity was used for cross-validation, where k-fold cross validation error was around 0.15. This method gave the minimum error, and the values of estimated eccentricity were unambiguous. Plot in Fig. C.3 depicts Root-Mean-Square Error (RMSE) for 20 different iterations obtained on randomly splitting the dataset into train–test sets:

Table C.15 tabulates few samples with imputed eccentricity values from the full training set (the complete catalog is available at astrirg.org/projects.html).

C.4. kNN imputation in detail

kNN imputation uses k Nearest Neighbors approach to impute values that are absent. For every observation to be imputed, it identifies k most similar observations based on the Euclidean distance and computes the weighted average (weight based on distance) of these k observations. The advantage is, being a lazy learning model, one could impute any or all the missing values in all attributes with one call to the function. We used the most frequent value among the k-nearest neighbors to estimate discrete attributes. The mean among the k-nearest neighbors is used to estimate values of continuous attributes (see Table C.16).

Table C.15
Comparison between known eccentricity values (P.Eccentricity) and the imputed ones, computed using kNN imputation. To ensure correctness (since imputation is an approximation procedure), we compare the imputed values with already known eccentricity values of the planets and verify that the error is within the claimed threshold RMSE of 0.15 (Fig. C.3).

Planet name	Imputed eccentricity	P. Eccentricity
55 Cnc e	0.0375	0.03
61 Vir b	0.15	0.12
CoRoT-7 b	0.15	0.12
EPIC 211822797 b	0.225	0.18
GJ 536 b	0.1	0.08
K2-3 d	0.0625	0.05
Kepler-23 b	0.075	0.06
Kepler-23 d	0.1	0.08
Kepler-296 e	0.125	0.1
WASP-47 e	0.0375	0.03

C.5. Algorithms used for imputation of eccentricity using kNN method

In this section, we have given various algorithms used to fill missing data of eccentricity of exoplanets. Algorithms 2 and 3 are described below. Algorithm 2 shows the steps used by kNN imputation method during training phase to estimate eccentricity values for the planets whose eccentricity value is marked as 0 and to compute the accuracy of the algorithm. Similarly, Algorithm 3 gives the steps used in testing phase of the algorithm.

Algorithm 2: Algorithm for kNN Imputation during training to report Accuracy.

```
Require: Samples with all values present. Split into Training and
  Testing sets. Testing set with features $i$, removed values $j$ stored
  in actual[$j$]. Training set with features $x$, feature $y$ which is to
  be imputed.
Ensure: This algorithm is repeated with different train–test folds
  and error is averaged out.
  for each pair $(i, j)$ in Testing set do
    for each pair $(x, y)$ in Training set do
      Calculate Euclidean distance $d \leftarrow d(i, x)$ and save set
  $S(x, y, d)$.
    end for
    Make set $T$ of $k$ smallest distances obtained.
    predicted[$j$] $\leftarrow$ mean($T[y]$)
  end for
  Compute RMSE taking the actual[j] and predicted[j] into
  consideration.
```

Algorithm 3: Algorithm for kNN Imputation during testing to estimate missing values.

```
Require: Testing set with features $i$, missing values $j$. Training set
  with features $x, y$ present.
  for each pair $(i, j)$ in Testing set do
    for each pair $(x, y)$ in Training set do
      Calculate Euclidean distance $d \leftarrow d(i, x)$ and save set
  $S(x, y, d)$.
    end for
    Make set T of K smallest distances obtained.
    Testing set[$j$] $\leftarrow$ mean($T[y]$)
  end for
```

Table C.16
CDHS scores of a sample of planets with and without imputed eccentricity.

Planet Name	Imputed Eccentricity	Imputed $CDHS_{CRS}$	Imputed $CDHS_{DRS}$	$CDHS_{CRS}$	$CDHS_{DRS}$
EPIC-206011691 b	0.1597	1.72	1.72	2.53	2.47
EPIC 212006344 b	0.2041	1.88	1.88	2.18	2.16
GJ 15 A b	0.1503	1.73	1.73	1.62	1.53
GJ 176 b	0.0986	1.98	1.98	1.82	1.71
Kepler-20 e	0.1281	1.84	1.93	0.89	0.98
Kepler-20 f	0.1460	1.52	1.52	1.01	1.01
Kepler-37 b	0.1717	1.37	1.54	0.68	0.87
Kepler-186 f	0.0400	1.15	1.15	1.09	1.08
Proxima Cen b	0.1944	1.09	1.09	1.09	1.08
TRAPPIST-1 e	0.1533	0.91	0.99	0.91	0.97
TRAPPIST-1 f	0.0780	0.94	1.02	0.98	0.98

Table C.17
CEESA scores as estimated by Particle Swarm Optimization (see Section 5); (a) under DRS constraint, and (b) under CRS constraint. r, d, t, υ, e, ρ and η are the parameters of Eq. (16), where η is assumed 1 under CRS constraint. Column *CEESA* records the maxima of the objective function Y, and i specifies the number of iterations taken to converge to the maximum.

Name	Class	r	d	t	υ	e	ρ	η	*CEESA*	i
GJ 176 b	non	0.304	0.001	0.375	0.271	0.050	0.467	0.808	1.52	85
GJ 667 C b	non	0.297	0.010	0.318	0.052	0.322	0.682	0.730	2.36	90
GJ 667 C e	psy	0.230	0.286	0.137	0.199	0.148	0.551	0.906	1.14	85
GJ 667 C f	psy	0.397	0.035	0.152	0.402	0.014	0.793	0.999	1.31	100
GJ 3634 b	non	0.178	0.175	0.005	0.194	0.447	0.894	0.657	2.07	94
HD 20794 c	non	0.073	0.142	0.452	0.190	0.144	0.953	0.635	1.20	78
HD 40307 e	non	0.156	0.307	0.185	0.033	0.319	0.428	0.939	2.69	88
HD 40307 f	non	0.272	0.231	0.064	0.305	0.127	0.676	0.802	1.28	77
HD 40307 g	psy	0.113	0.219	0.066	0.454	0.148	0.711	0.991	3.26	92
Kepler-186 f	hyp	0.039	0.159	0.116	0.329	0.357	0.253	0.919	1.35	70
Proxima Cen b	psy	0.272	0.173	0.284	0.193	0.079	0.615	0.114	0.99	75
TRAPPIST-1 b	non	0.488	0.151	0.039	0.193	0.129	0.151	0.014	0.99	87
TRAPPIST-1 c	non	0.172	0.236	0.275	0.242	0.075	0.969	0.962	1.06	80
TRAPPIST-1 d	mes	0.106	0.308	0.075	0.218	0.293	0.844	0.017	0.99	93
TRAPPIST-1 e	psy	0.189	0.266	0.192	0.094	0.260	0.371	0.006	0.99	84
TRAPPIST-1 g	hyp	0.326	0.186	0.143	0.278	0.067	0.315	0.021	1.00	76

(a) Estimated habitability scores by CEESA under DRS constraint.

Name	Class	r	d	t	υ	e	ρ	η	*CEESA*	i
GJ 176 b	non	0.194	0.020	0.315	0.465	0.006	0.398	1.000	1.88	86
GJ 667 C b	non	0.162	0.289	0.090	0.087	0.372	0.836	1.000	3.54	107
GJ 667 C e	psy	0.373	0.032	0.134	0.304	0.157	0.217	1.000	1.25	71
GJ 667 C f	psy	0.394	0.006	0.043	0.360	0.196	0.490	1.000	1.44	81
GJ 3634 b	non	0.351	0.122	0.006	0.069	0.453	0.439	1.000	2.89	96
HD 20794 c	non	0.101	0.077	0.691	0.071	0.059	0.756	1.000	1.58	94
HD 40307 e	non	0.069	0.091	0.097	0.173	0.569	0.768	1.000	5.29	94
HD 40307 f	non	0.285	0.161	0.053	0.443	0.058	0.342	1.000	1.42	73
HD 40307 g	psy	0.156	0.010	0.081	0.302	0.451	0.612	1.000	7.15	94
Kepler-186 f	hyp	0.036	0.017	0.082	0.383	0.483	0.929	1.000	1.68	85
Proxima Cen b	psy	0.352	0.383	0.103	0.059	0.103	0.936	1.000	0.89	83
TRAPPIST-1 b	non	0.148	0.147	0.344	0.269	0.093	0.767	1.000	0.94	81
TRAPPIST-1 c	non	0.038	0.060	0.575	0.321	0.005	0.602	1.000	1.17	86
TRAPPIST-1 d	mes	0.023	0.065	0.475	0.391	0.045	0.830	1.000	0.84	79
TRAPPIST-1 e	psy	0.176	0.464	0.253	0.103	0.004	0.920	1.000	0.86	81
TRAPPIST-1 g	hyp	0.060	0.086	0.310	0.540	0.004	0.848	1.000	0.97	86

(b) Estimated habitability scores by CEESA under CRS constraint.

Tables C.17a and C.17b show the CEESA scores for some of the exoplanets estimated using Particle Swarm Optimization. Column *CEESA* shows CEESA score and column *Class* shows the habitability class of each planet.

Appendix D. Parameters used in fuzzy ANN classification method

There are 68 parameters in the PHL-EC dataset. Not all of them are important in classification of the planets. 45 relevant features were found out for classification. Table D.18 shows the list of parameters used for fuzzy classification in Case 1 to Case 4 in Section 6.4.

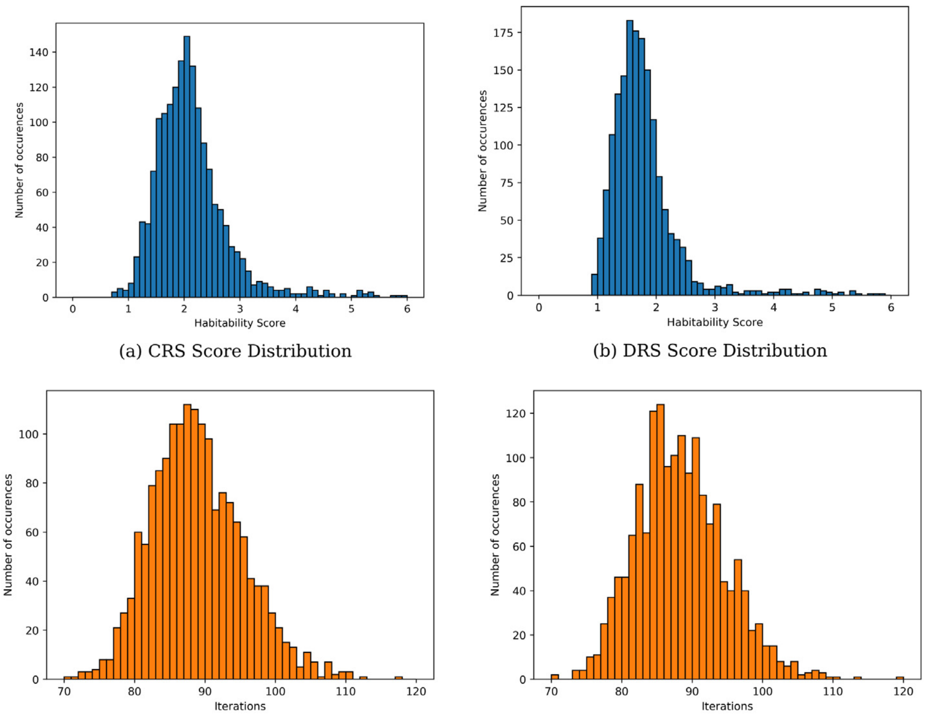
Appendix E. MATLAB codes

Here we present Matlab codes that implement the analytical model, compute the scores for the entire dataset.

E.1. Function fmincon

The function *fmincon* finds a constrained minimum of a scalar function of multivariable starting at an initial point. This is generally known as constrained nonlinear optimization. Function *fmincon* solves problems of the form: $\min f(x)$ subject to x, where

$$\begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \end{cases}$$



(a) CRS Score Distribution

(b) DRS Score Distribution

(c) CRS Iterations Distribution

(d) DRS Iterations Distribution

Fig. F.4. Plots for the Constant Elasticity Earth Similarity Approach.

Table D.18
Parameters used in fuzzy classification for 3-class dataset in Case 1.

Sl.No.	Parameter name	Sl.No.	Parameter name
1	P. Zone Class	27	P. Inclination (deg)
2	P. Mass Class	28	P. Omega (deg)
3	P. Composition Class	29	S. Mass (SU)
4	P. Atmosphere Class	30	S. Radius (SU)
5	P. Min Mass (EU)	31	S. Teff (K)
6	P. Mass (EU)	32	S. Luminosity (SU)
7	P. Radius (EU)	33	S. [Fe/H]
8	P. Density (EU)	34	S. Age (Gyrs)
9	P.Gravity (EU)	35	S. Appar Mag
10	P. Esc Vel (EU)	36	S. Mag from Planet
11	P. SFlux Min (EU)	37	S. Size from Planet (deg)
12	P. SFlux Mean (EU)	38	S. Hab Zone Min (AU)
13	P. SFlux Max (EU)	39	S. Hab Zone Max (AU)
14	P. Teq Min (K)	40	P. HZD
15	P. Teq Mean (K)	41	P. HZC
16	P. Teq Max (K)	42	P. HZA
17	P. Ts Min (K)	43	P. HZI
18	P. Ts Mean (K)	44	P. ESI
19	P. Ts Max (K)	45	P. Habitable
20	P. Surf Press (EU)		
21	P. Mag		
22	P. Appar Size (deg)		
23	P. Period (days)		
24	P. Sem Major Axis (AU)		
25	P. Eccentricity		
26	P. Mean Distance		

are the linear constraints, and the following equations are the non-linear constraints,

$$\begin{cases} Cx \leq 0 \\ C_{eq}x = 0 \end{cases}$$

with bounding of variables:

$$\begin{cases} lb \leq x \\ x \leq ub\,. \end{cases}$$

This has been applied to **CRS** and **DRS** cases for the CEESA and CES scores computation.

E.2. Constant returns to scale

Apply the constraints

$$\begin{cases} a+b+c+d+e=1 \\ \rho \leq 1,\ \nu = 1 \end{cases}$$

to the function $Y = (a.x_1^\rho + b.x_2^\rho + c.x_3^\rho + d.x_4^\rho + e.x_5^\rho)^{\nu/\rho}$, use *fmincon* to compute ρ and ν for the optimum Y.

E.3. Decreasing returns to scale

Apply the constraints

$$\begin{cases} a+b+c+d+e=1 \\ \rho \leq 1,\ \nu < 1 \end{cases}$$

to the function $Y = \left(a.x_1^\rho + b.x_2^\rho + c.x_3^\rho + d.x_4^\rho + e.x_5^\rho\right)^{\nu/\rho}$, use *fmincon* to compute ρ and ν for the optimum Y.

E.4. Implementation of fmincon

$[x, fval]$ = fmincon(fun, x_0, A, b) starts at point x_0 and finds a minimum x to the function described in fun subject to the linear inequalities, $A{*}x \leq b$, where A is a matrix, x and b are vectors and x_0 can be a scalar, a vector or a matrix. It also returns the value of the objective function **fun** at the solution x.

$[x, fval]$ = fmincon(fun, x_0, A, b, A_{eq}, b_{eq}) starts at x_0 and minimizes **fun** subject to the linear inequalities $A_{eq} * x = b_{eq}$ and

$A * x \leq b$, where A_{eq} is a matrix and b_{eq} is a vector. It also returns the value of the objective function **fun** at the solution x.

$[x, fval] = \text{fmincon}(\text{fun}, x_0, A, b, A_{eq}, b_{eq}, lb, ub)$ defines a set of lower and upper bounds on the design variables in x, so that the solution is always in the range $lb \leq x \leq ub$. If no equalities exist, set $Aeq = []$ and $beq = []$. If $x(i)$ is unbounded below, set $lb(i) = -\text{Inf}$, and if $x(i)$ is unbounded above, set $ub(i) = \text{Inf}$ (Documentation, 2017).

Appendix F. Additional information on results

Plots in Figs. F.4(a) and F.4(b) describe the distribution of CEESA scores across the exoplanets, while plots in Figs. F.4(c) and F.4(d) show the distribution of iterations to convergence. These figures aggregate the results of optimizing the habitability production functions for each exoplanet in the PHL-EC using method described in Algorithm 1.

References

Arrow, K.J., Chenery, H.B., Minhas, B.S., Solow, R.M., 1961. Capital-labor substitution and economic efficiency. Rev. Econ. Stat. 43 (225), doi:10.2307/1927286.

Batalha, N.M., 2014. Exploring exoplanet populations with NASA's Kepler Mission. Proc. Natl. Acad. Sci. 111 (12647).

Bora, K., Saha, S., Safonova, M., Routh, S., Narasimhamurthy, A.M., 2016. CD-HPF: new habitability score via data analytic modeling. Astron. Comput. 17, 129–143.

Cassan, A., Kubas, D., Beaulieu, J.-P., et al., 2012. One or more bound planets per Milky Way star from microlensing observations. Nature 481 (167).

Cobb, C.W., Douglas, P.H., 1928. A theory of production. Amer. Econ. Rev. 18 (Supplement), 139.

Dayal, P., Cockell, C., Rice, K., Mazumdar, A., 2015. The quest for cradles of life: using the fundamental metallicity relation to hunt for the most habitable type of galaxy. Astrophys. J. Lett. 810 (L2).

Documentation on fmincon function. https://in.mathworks.com. Retrieved on 12/04/2017.

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: IEEE Proc. Sixth International Symposium on Micro Machine and Human Science, vol. 39. doi:10.1109/MHS.1995.494215.

van Elteren, A., Portegies, Zwart S., Pelupessy, I., Cai, M.X., McMillan, S.L.W., 2019. Survivability of planetary systems in young and dense star clusters. Astron. Astrophys. 624 (A120).

Ginde, G., Saha, S., Mathur, A., Venkatagiri, S., Vadakkepat, S., Narasimhamurthy, A., Daya Sagar, B.S., 2016. ScientoBASE: A Framework and model for computing scholastic indicators of non-local influence of journals via native data acquisition algorithms. J. Scientometrics 107 (1), 1–51.

Gonzalez, G., Brownlee, D., Ward, P., 2001. The galactic habitable zone: Galactic chemical evolution. Icarus 152.

Hájková, D., Hurnik, J., 2007. Cobb–Douglas: the case of a converging economy. Czech J. Econ. Finance (Finance a uver) 57, 465.

Hardy, G.H., Littlewood, J.E., et al., 1952. Inequalities. Cambridge University Press, pp. 1–324.

Hassani, A., 2012. Applications of cobb–douglas production function in construction time-cost analysis (M.Sc. thesis). University of Nebraska, Lincoln.

Heller, R., Armstrong, J., 2014. Superhabitable worlds. Astrobiology 14 (50).

Hossain, M., Majumder, A., Basak, T., 2012. An application of non-linear Cobb–Douglas production function to selected manufacturing industries in Bangladesh. Open J. Stat. 2 (460), doi:10.4236/ojs.2012.24058.

Huang, S.-S., 1959. The problem of life in the universe and the mode of star formation. Publ. Astron. Soc. Pac. 71 (421).

Irwin, L.N., Méndez, A., Fairén, A.G., Schulze-Makuch, D., 2014. Assessing the possibility of biological complexity on other worlds, with an estimate of the occurrence of complex life in the Milky Way Galaxy. Challenges 5 (159).

Irwin, L.N., Schulze-Makuch, D., 2011. Cosmic Biology: How Life Could Evolve on Other World. Springer-Praxis, New York.

Kaltenegger, L., Udry, S., Pepe, F., 2011. A habitable planet around HD 85512. Preprint, arXiv:1108.3561.

Kasting, J.F., 1993. Earth's early atmosphere. Science 259 (920).

Limbach, M.A., Turner, E.L., 2015. Exoplanet orbital eccentricity: multiplicity relation and the solar system. Proc. Natl. Acad. Sci. 112 (20).

Luhman, K.L., Burgasser, A.J., Bochanski, J.J., 2011. Discovery of a candidate for the coolest known brown dwarf. Astrophys. J. Lett. 730, L9. doi:10.1088/2041-8205/730/1/L9.

Méndez, A., 2011. A Thermal Planetary Habitability Classification for Exoplanets. Planetary Habitability Laboratory @ UPR Arecibo, URL: http://phl.upr.edu/library/notes/athermalplanetaryhabitabilityclassificationforexoplanets.

Méndez, A., Rivera-Valentin, E.G., 2017. The equilibrium temperature of planets in elliptical orbits. Astrophys. J. Lett. 837 (L1).

Nemirovski, A.S., Todd, M.J., 2008. Interior-point methods for optimization. Acta Numer. 17 (191), doi:10.1017/S0962492906370018.

Powers, D.M.W., 2011. Evaluation: From precision, recall and F-factor to ROC, informedness, markedness and correlation. J. Mach. Learn. Technol. 2, 37–63.

Ray, T., Liew, K.M., 2001. A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimisation problems. In: Proc. 2001 Congress on Evolutionary Computation, vol. 1, pp. 75–80.

Ricardo, P., 2008. Analysis of the publications on the applications of particle swarm optimisation. J. Artif. Evol. Appl. Article 4 (2008), 10. doi:10.1155/2008/685175.

Safonova, M., Murthy, J., Shchekinov, Y.A., 2016. Age aspects of habitability. Int. J. Astrobiol. 15 (93).

Saha, S., Basak, S., Safonova, M., Bora, K., Agrawal, S., Sarkar, P., Murthy, J., 2018a. Theoretical validation of potential habitability via analytical and boosted tree methods: An optimistic study on recently discovered exoplanets. Astron. Comput. 23, 141–150.

Saha, S., Bora, K., Basak, S., Mathur, A., Agrawal, S., 2018b. Habitability classification of exoplanets: a machine learning insight. arXiv:1805.08810.

Saha, S., Bora, K., Mathur, A., Basak, S., Agrawal, S., 2018c. Saha-bora activation function: habitability classification. Preprint, doi:10.13140/RG.2.2.21081.62565.

Saha, S., Sarkar, J., Dwivedi, A., Dwivedi, N., Narasimhamurthy, A.M., Roy, R., 2016. A novel revenue optimization model to address the operation and maintenance cost of a data center. J. Cloud Comput. Adv. Syst. Appl. 5, 1. doi:10.1186/s13677-015-0050-8.

Schulze-Makuch, D., Méndez, A., Fairén, A.G., et al., 2011. A two-tiered approach to assessing the habitability of exoplanets. Astrobiology 11 (1041).

Shi, Y., Eberhart, R., 1998. A modified particle swarm optimizer. In: IEEE World Congress on Computational Intelligence, Proc. The 1998 IEEE International Conference on Evolutionary Computation, vol. 6, pp. 9–73.

Stevenson, D.J., 1999. Life-sustaining planets in interstellar space? Nature 400 (6739), 32.

Strigari, L.E., Barnabè, M., Marshall, P.J., Blandford, R.D., 2012. Nomads of the galaxy. Mon. Not. R. Astron. Soc. 423 (1856).

Wang, Y., Liu, Y., Tian, F., Hu, Y., Huang, Y., 2017. Effects of eccentricity on climates and habitability of terrestrial exoplanets around m dwarfs. Preprint, arXiv:1710.01405.

Wittenmyer, R.A., Tuomi, M., Butler, R.P., et al., 2014. GJ 832c: A super-earth in the habitable zone. Astrophys. J. 791 (114).

Wolf, E.T., 2017. Assessing the habitability of the TRAPPIST-1 system using a 3D climate model. Astrophys. J. 839 (L1).

Wu, D.-M., 1975. Estimation of the Cobb–Douglas production function. Econometrica 43 (739), doi:10.2307/1913082.

Zadeh, L.A., 1965. Fuzzy sets. Inf. Control 8, 338. doi:10.1016/S0019-9958(65)90241-X.