



## OPEN ACCESS

## EDITED BY

Josep M. Trigo-Rodríguez,  
Spanish National Research Council  
(CSIC), Spain

## REVIEWED BY

Jianguo Yan,  
Wuhan University, China  
Thomas Burbine,  
Mount Holyoke College, United States

## \*CORRESPONDENCE

M. V. F. Lourenço,  
m.lourenco@unesp.br

## SPECIALTY SECTION

This article was submitted to Planetary  
Science,  
a section of the journal  
Frontiers in Astronomy and Space  
Sciences

RECEIVED 07 July 2022

ACCEPTED 16 August 2022

PUBLISHED 08 September 2022

## CITATION

Lourenço MVF and Carruba V (2022),  
Genetic optimization of asteroid  
families' membership.  
*Front. Astron. Space Sci.* 9:988729.  
doi: 10.3389/fspas.2022.988729

## COPYRIGHT

© 2022 Lourenço and Carruba. This is  
an open-access article distributed  
under the terms of the [Creative  
Commons Attribution License \(CC BY\)](#).  
The use, distribution or reproduction in  
other forums is permitted, provided the  
original author(s) and the copyright  
owner(s) are credited and that the  
original publication in this journal is  
cited, in accordance with accepted  
academic practice. No use, distribution  
or reproduction is permitted which does  
not comply with these terms.

# Genetic optimization of asteroid families' membership

M. V. F. Lourenço\* and V. Carruba

School of Natural Sciences and Engineering, São Paulo State University (UNESP), São Paulo, Brazil

Asteroid families are groups of asteroids with a common origin, such as prior collisions or the parent body's rotational fission. In proper  $[a, e, \sin(i)]$  element domains, they are generally observed using the hierarchical clustering technique (HCMs), but the method may be ineffective in high-density regions, where it may be unable to separate near families. Previous works employed a different technique in which nine different machine learning classification algorithms were applied to the orbital distribution in proper elements of 21 known family constituents for the goal of new members' identification. Each algorithm's optimal hyper-parameters for every family were extensively investigated, which proved to be a time-consuming and repetitive procedure. Herein, we used a genetic algorithm-based tool to identify the most optimal machine learning algorithm for the same studied asteroid families as an alternative to the originally utilized parameter search mode. When compared to the same evaluative metrics utilized in the previous machine learning application study, the precision values of the new genetic machine learning algorithms have been consistently comparable, demonstrating that this alternative technique can be satisfactorily efficient and fast.

## KEYWORDS

methods data analysis, methods statistical, minor planets and asteroids: general, minor planets, methods, estimation

## 1 Introduction

Asteroid families are groups of minor planets that can be identified in domains of proper elements  $[a, e, \sin(i)]$ , with  $a$  being the asteroid's proper semi-major axis,  $e$  the proper eccentricity, and  $i$  the proper inclination. Contrary to osculating elements that change on timescales of days or less, proper elements are constant in motion and are stable on timescales of millions of years when non-gravitational forces are not considered. Proper elements can be obtained either through perturbation theory approaches or, most commonly, through numerical methods involving N-body simulations of the planets and asteroids' orbits and Fourier analysis (Knežević and Milani, 2003). The latter elements are called synthetic proper elements. Since the 90s, the most commonly used method for detecting asteroid families has been the hierarchical clustering method (HCM, hereafter, Zappalà et al. (1995); see also Bendjoya and Zappalà (2002) for a later review). This approach has been very successful in detecting new asteroid families and is the current standard for recent surveys, like those conducted by Nesvorný et al. (2015), and for the

**TABLE 1** Table of the main belt orbital regions. We report the region name, where H.I. indicates for highly inclined, the  $a_{\min}$  value and the Jupiter mean-motion resonance name associated with that value,  $a_{\max}$  value and its related Jupiter mean-motion resonance name, and the minimum value of  $\sin(i_{\min})$ .

Region name	$a_{\min}$ (au)	Res. name	$a_{\max}$ .(au)	Res. name	$\sin(i_{\min})$
Hungaria region	1.780	-	2.253	7J:2A	0.000
H.I. inner main belt	2.257	7J:2A	2.465	3J:1A	0.300
H.I. central main belt	2.520	3J:1A	2.818	5J:2A	0.300
H.I. outer main belt	2.832	5J:2A	3.240	2J:1A	0.300
Cybele region	3.290	2J:1A	3.800	5J:3A	0.000

**TABLE 2** Extratree and random forest classifier machine learning parameters as generated by the *TPOT* tool, with the number of estimators fixed at 100. The columns report the asteroid families IDs, the values of bootstrap, criterion, max\_features, min\_sample\_leaf, and min\_sample\_split.

Family	Bootstrap	Criterion	Max_features	Min_sample_leaf	Min_sample_split
170 Maria	True	Gini	0.65	3	2
847 Agnia	True	Gini	0.4	3	7
363 Padua	False	Gini	0.2	1	7
410 Chloris	False	Entropy	0.5	3	10
569 Misa	False	Entropy	0.9	2	3
221 Eos	False	Gini	0.9	3	15
24 Themis	True	Entropy	0.5	1	4
158 Koronis	False	Entropy	0.3	1	4
10 Hygiea	False	Gini	0.4	3	7
375 Ursula	True	Gini	0.5	1	4
1040 Klumpkea	False	Entropy	0.45	1	15
283 Emma	False	Entropy	1	2	11
845 Naema	False	Entropy	0.7	3	17

families available at the Asteroid Families Portal (AFP, <http://asteroids.matf.bg.ac.rs/fam/index.php>, Radović et al. (2017)). As discussed in Milani et al. (2014), however, the large number density of asteroids in some regions of the main belt may introduce a problem not present in the smallest databases used in the 90s. Nearby asteroid families may overlap and be very close in proper element domains. As an effect, traditional HCM may fail to recognize these groups as separate. This problem, referred to as *chaining* by Milani et al. (2014), may be overcome if machine learning (ML) methods are employed. Carruba et al. (2020) used supervised ML algorithms to identify new members of known asteroid families, as listed at the AFP. The basic idea behind these approaches was to train ML methods to recognize patterns among the  $[a, e, \sin(i)]$  distribution of already known family members, so that new asteroids with similar values of  $[a, e, \sin(i)]$  could be attached to these groups without the issue of *chaining*. One limitation of that approach was that those algorithms were dependent on free hyper-parameters, the optimal values of which had to be found

by searching over extended phase space. This was a time-consuming endeavor. Genetic algorithms, as described in Section 2.3, allow for the selection of the best-performing method and of its most optimal values of hyper-parameters automatically. The goal of this work is to verify if their use could be a safe and viable option.

## 2 Methods

### 2.1 The use of ML methods for asteroid family identification: Some background

In 2020, the AFP site reported data for asteroid families identified among a database of 631,226 numbered asteroids using the hierarchical clustering method [HCM, Bendjoya and Zappalà (2002)]; 21 asteroid families from the AFP were used to train the (ML) algorithms by dividing the samples into two parts: one for asteroids with an absolute magnitude  $H$  lower than 14, the

TABLE 3 Table reports the metrics of the 21 asteroid families from the Asteroid Families Portal using provided genetic machine learning algorithm by TPOT. There is a representation of the metrics  $f_1$  (completeness), precision, the FP coefficients, and the best estimator of each family.

Family ID	$f_1$	Precision	FP	Best estimator
20 Massalia	0.21	0.85	0.62	Gradient boosting classifier
163 Erigone	0.21	0.90	0.65	KNN
15 Eunomia	0.15	0.69	0.50	XGBoost
170 Maria	0.19	0.76	0.55	Random forest classifier
668 Dora	0.20	0.62	0.60	XGBoost
847 Agnia	0.19	0.83	0.60	Random forest classifier
363 Padua	0.12	0.77	0.55	Random forest classifier
1726 Hoffmeister	0.23	0.83	0.61	XGBoost
410 Chloris	0.22	0.95	0.69	Random forest classifier
808 Merxia	0.23	0.82	0.60	XGBoost
128 Nemesis	0.17	0.74	0.54	XGBoost
569 Misa	0.24	0.82	0.61	Extra tree
221 Eos	0.22	0.78	0.57	Extra tree
24 Themis	0.15	0.85	0.61	Random forest classifier
158 Koronis	0.27	0.83	0.61	Extra tree
10 Hygiea	0.16	0.85	0.61	Random forest classifier
375 Ursula	0.27	0.80	0.59	Random forest classifier
1040 Klumpkea	0.30	0.64	0.50	Random forest classifier
283 Emma	0.24	0.83	0.61	Extra tree
845 Naema	0.21	0.70	0.52	Extra tree
490 Veritas	0.27	0.79	0.59	XGboost

training sample, and one for the rest, the test sample. This value of  $H$  was justified based on the work of Milani et al. (2014), which showed that asteroid families obtained for samples with  $H < 14$  are not affected by the chaining issue and should present minimal or no overlap with other nearby dynamical families.

These families have to satisfy the criteria of being available in the Radović et al. (2017) catalog and of having at least 10 members with  $H < 14$ . The outcome of ML algorithms was measured based on metrics defined using the similarity matrix. As in the study by Carruba et al. (2019), one can define true positive (TPos) asteroids identified as a family member by both the HCM procedure method and the machine learning application, false positive (FPoS) as the ones identified only by machine learning, and the false negatives (FNeg) as asteroids identified by HCM alone. Based on these quantities,  $f_1$  or “completeness,” “precision,” and the final parameter FP metrics can be introduced. Completeness is defined as follows:

$$f_1 = \text{Completeness} = \frac{TPos}{TPos + FNeg} = \frac{TPos}{N_{Or}} \quad (1)$$

This formula alone is not a good parameter to measure the algorithm’s performance. For example, if we have a small original family with 100 members and a retrieved family of 10,000 that

TABLE 4 Table with FP coefficient metric comparison results of Carruba et al. (2020) and the current work. We report the asteroid families’ names, the value of the result metric FP of 2020, the current results of the FP, and a percentual comparison R (see Eq. 4).

Family id	FP 2020	FP 2022	R
20 Massalia	0.64	0.62	-0.032
163 Erigone	0.78	0.65	-0.2
15 Eunomia	0.89	0.50	-0.78
170 Maria	0.85	0.55	-0.55
668 Dora	0.96	0.60	-0.6
847 Agnia	0.86	0.60	-0.43
363 Padua	0.74	0.55	-0.35
1726 Hoffmeister	0.89	0.61	-0.46
410 Chloris	0.87	0.69	-0.26
808 Merxia	0.90	0.60	-0.5
128 Nemesis	0.81	0.54	-0.5
569 Misa	0.81	0.61	-0.33
221 Eos	0.89	0.57	-0.56
24 Themis	0.95	0.61	-0.56
158 Koronis	0.96	0.61	-0.57
10 Hygiea	0.93	0.61	-0.52
375 Ursula	0.85	0.59	-0.44
1040 Klumpkea	0.91	0.50	-0.82
283 Emma	0.89	0.61	-0.46
845 Naema	0.94	0.52	-0.81
490 Veritas	0.94	0.59	-0.59

includes all the 100 original members, the result of completeness would be maximum. The retrieved family is not a good approximation of the original one. To address this problem, the precision formula was defined as follows:

$$\text{Precision} = \frac{TPos}{TPos + FPos} = \frac{TPos}{N_{Retr}} \quad (2)$$

where  $N_{Retr} = TPos + FPos$  and is the number of asteroids in the retrieved family. High precision values can also be associated with a small family; for example, if we have an original family of 100 members and a retrieved one composed of 10 TPos objects, the formula outcome will be 1, but again, the retrieved family would not be a good representation of the original.

The two parameters can be combined into a single optimized measure through the final parameter formula:

$$FP = \frac{1}{\sqrt{2}} \sqrt{\text{Completeness}^2 + \text{Precision}^2} \quad (3)$$

## 2.2 Machine learning methods

Current artificial intelligence research places a strong emphasis on the machine learning field, and a variety of ML

TABLE 5 Table shows the *Fp*'s metrics percentual comparison *R* between Carruba et al. (2020) machine learning algorithms applied to an updated database (displayed in the second column) and the current machine learning algorithms generated by genetic algorithms (column FP 2022).

Family ID	FP 2020 (new)	FP 2022	R
20 Massalia	0.629	0.621	-0.013
163 Erigone	0.620	0.650	0.046
15 Eunomia	0.598	0.532	-0.124
170 Maria	0.477	0.555	0.141
668 Dora	0.612	0.598	-0.023
847 Agnia	0.610	0.605	-0.008
363 Padua	0.558	0.553	-0.009
1726 Hoffmeister	0.606	0.607	0.002
410 Chloris	0.649	0.690	0.059
808 Merxia	0.530	0.600	0.116
128 Nemesis	0.529	0.539	0.019
569 Misa	0.548	0.607	0.097
221 Eos	0.600	0.574	-0.045
24 Themis	0.610	0.610	0.000
158 Koronis	0.612	0.614	0.003
10 Hygiea	0.609	0.612	0.005
375 Ursula	0.612	0.598	-0.023
1040 Klumpkea	0.507	0.498	-0.018
283 Emma	0.579	0.621	0.068
845 Naema	0.534	0.520	-0.027
490 Veritas	0.606	0.559	-0.084

algorithm methods have been developed. Decision tree-based bagging and boosting are the most popular and widely used principles of ML algorithm codes.

Decision tree is a simple machine learning algorithm that makes decisions in the form of a tree, with branches representing decision rules and leaf nodes representing results. Extra tree uses decision trees to produce the best results by using the standalone majority vote method.

The bagging boosting principle is followed by the bagging classifier algorithm, in which the training sample is divided into multiple samples, called bootstrap, and each one is used to train an independent classifier, resulting in a variety of predictions. The final result is a weighted average of the results of each classifier.

Adaptive boosting, such as gradient methods, uses the ensemble method to combine the results of multiple standalone classifiers to arrive at the best result. However, boosting methods use the concept of the least accurate prediction getting lower weights, and the final result also follows the bagging classifier idea. Adaptive boosting (AdaBoost) trains the classifier over the data points first, and the ones that were incorrectly identified are given a higher weight, influencing the outcome of the following interaction.

When there are no significant errors or the number of iterations is reached, the process stops.

The gradient boosting (Gboost) algorithm applies the gradient concept to the bagging weights idea, in which the method finds under-performing estimators and attempts to fit and correct them in the next iteration by adding new estimators. When no further improvements are conceivable, the process ends.

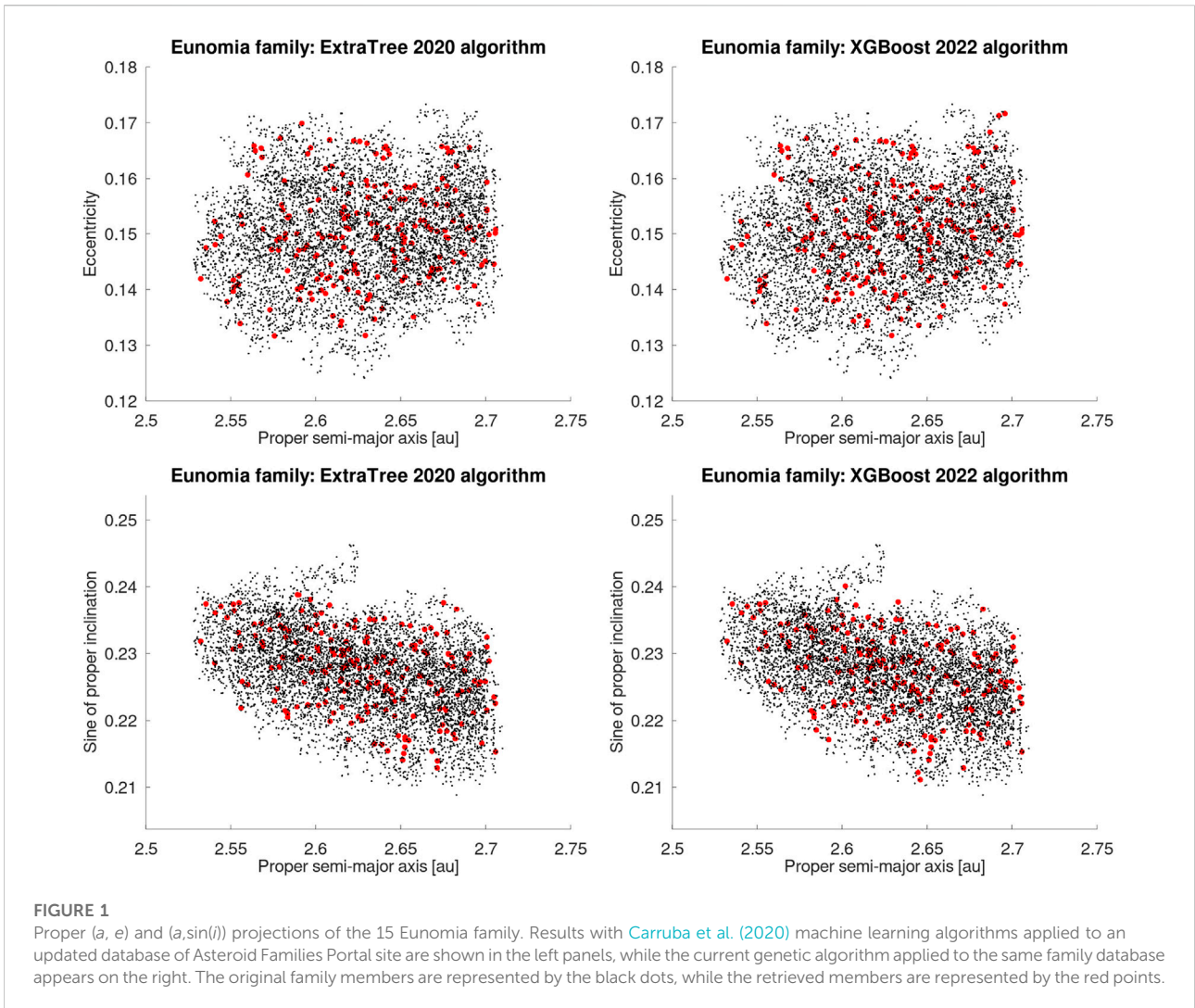
The eXtreme gradient boosting (XGboost) technique is similar to Gboost, but it offers speed improvements such as greater multicore support, processor cache use, and the ability to control over-fitting using a more regularized model formalization.

The non-parametric supervised learning technique K-nearest neighbor (KNN) method is one of the simplest machine learning algorithms. This simple method is based on the study of the uncatalogued data's neighbors. When the algorithm receives unseen data, it examines the classes of its closest neighbors to predict the new data category. For example, if there is an unknown color ball (data) between others with well-defined color (classes) balls, such as three reds and two blues, if we just judge by the colors of the balls, the unknown data can be identified as red due to the actual dominating class (the larger number of neighbors with the same class). This principle may be applied to any type of data and can be used to predict the class of a point by looking at its surroundings.

### 2.3 Genetic algorithms

Genetics algorithms, initially developed by Bremermann (1958) and popularized by Holland (1962), are a search and optimization programming technique based on Darwin's theory of species evolution, where the strongest individual is favored and his reproduction is more likely than those of the others, forming a new generation.

In terms of the genetic algorithm, each individual in the population is a solution, and every individual's collection of hyper-parameters (genes) determines this solution's characteristics when we have a problem to solve. A genetic algorithm's functionality can be divided into five sections: population, fitness function, selection, crossover, and mutation. The fitness function assigns a fitness score to each solution, indicating the individual's capacity to compete with the others. The fittest solutions are chosen during the selection phase, and their characteristics are passed down to the next generation. Each pair of parents is mated, and a crossover point is picked at random from the available genes. The offspring is formed by exchanging the available genes until the stipulated crossover point is reached, and then the newly formed individual is included in the population. The crossover is defined by this phase. At the mutation stage, newly produced offspring may be subjected to a low-probability random mutation, altering their



genetic structure. The algorithm determines if the new population is considerably different from the previous generation at the termination phase and if the genetic algorithm supplied a set of solutions to our problem. If not, a new circle begins.

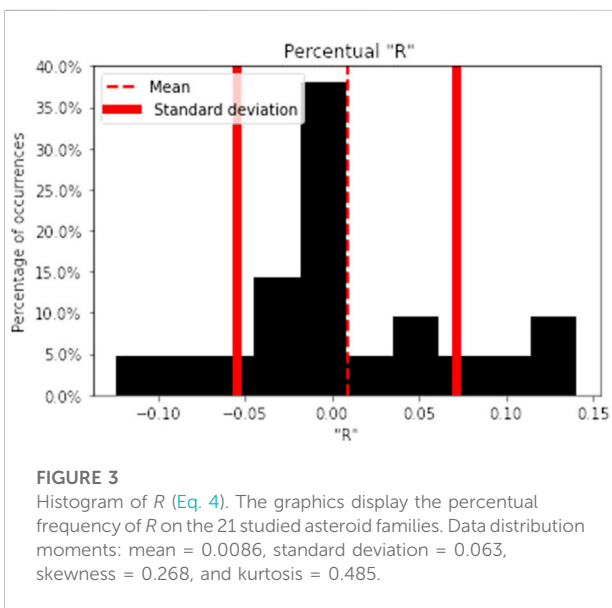
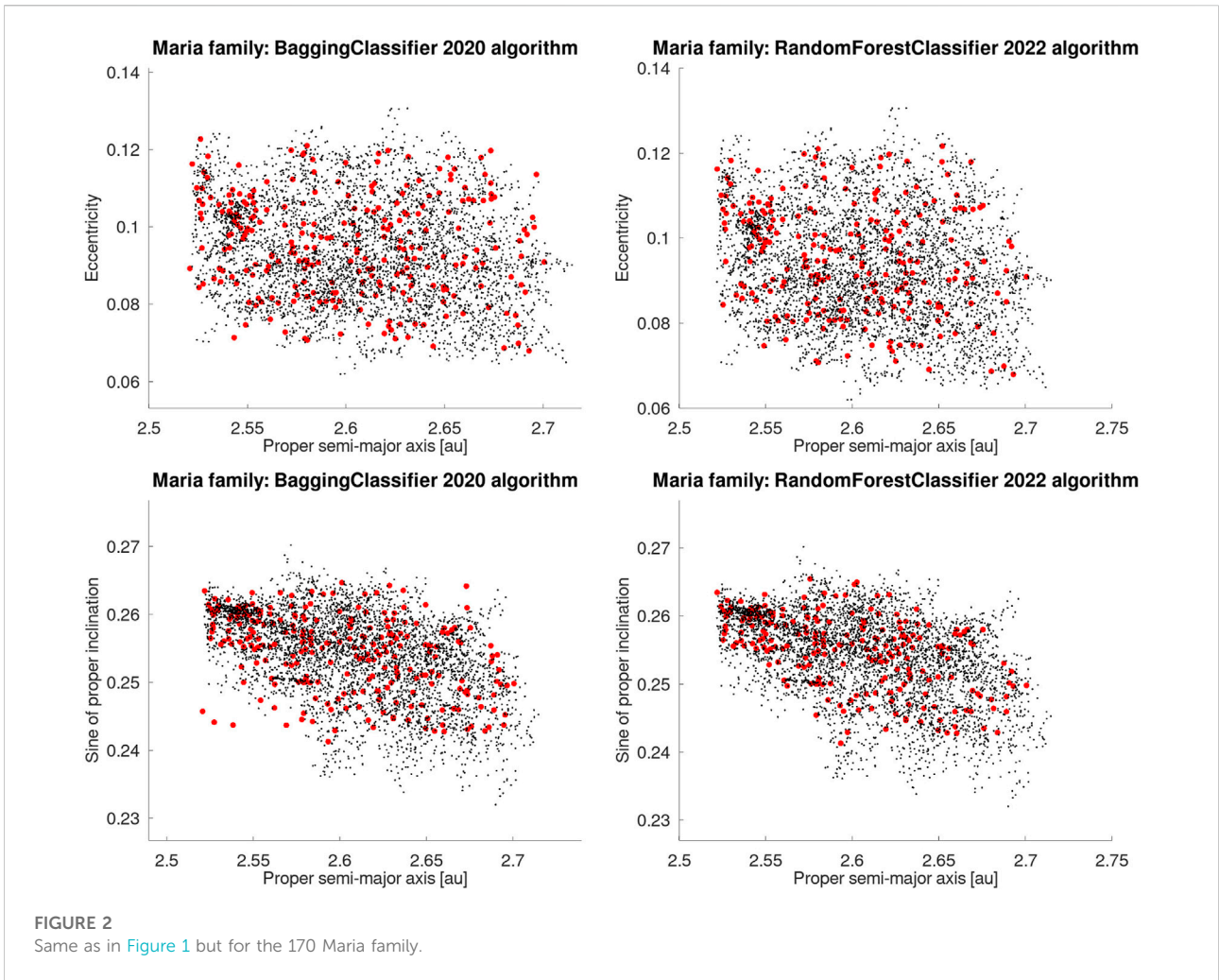
*TPOT* (Le et al., 2020; Olson et al., 2016a; Olson et al., 2016b) is a Python-based automated machine learning tool that employs genetic algorithms to identify the most appropriate machine learning pipeline for a particular task. The user must provide simple inputs after manually cleaning the raw user data: generations (number of genetic algorithm training iterations), population size (number of individuals retained in the population each generation), cross-validation—*cv* (used to evaluate each pipeline using a simple parameter  $K$ , which corresponds to the number of groups into which the data sample is divided), and the random state (random number generator seed for reproducibility).

The tool then automatically produces the machine learning pipeline’s best model.

### 2.4 Procedure

The main goal of the genetic algorithm is to find a minimum in the fitness function. *TPOT* may not always find a global minimum of this function. This means that, occasionally, the machine learning pipeline produced by *TPOT* may not be the best-performing one. In this work, we estimate the efficiency of using genetic algorithms compared to other methods of optimization of hyper-parameters like those used in Carruba et al. (2020).

The main criteria for a family to be included in this research are that 1) it has to be listed in the Radović et al. (2017) catalog and 2) it has to have at least 10 members with  $H < 14$  and be



located in the main belt regions (Carruba et al., 2013); 21 asteroid families satisfy our selection criteria on the Asteroid Families Portal (AFP). For these families, we first selected the region where the group is located (Table 1). We then label asteroids in the region with 1 if the asteroid belongs to a given family of interest, and 0 otherwise. *TPOT* is then used to train on this set of elements, yielding the optimum machine learning pipeline for this family.

### 3 Results

The main objective of the research is to investigate the use of genetic algorithms to optimize machine learning predictions of new asteroid family members as an improvement to the work carried out by Carruba et al. (2020). The machine learning algorithm parameters generated by *TPOT* are listed in Table 2 for extra tree and random forest classifier methods. The other families' algorithm parameters (XGBoost, gradient boosting

classifier, and K neighbors classifier) are available upon request. Table 3 shows the new metrics values, as well as the best TPOT, generated machine learning algorithm for each family<sup>1</sup>.

Carruba et al. (2020) used machine learning parameter algorithms applied to an old database of numbered asteroids at the Asteroid Families Portal, making a direct percentual comparison (Eq. 4) to current results, which are based on an updated asteroid database, complicated. Previously, HCM was applied to a database of 631,226 numbered asteroids. There were 793,310 registered objects in May 2022. Changes between the FP's (Eq. (3)) coefficient metrics obtained in Carruba et al. (2020) and the current ones are displayed in Table 4.

$$R = \frac{(FP_{2022}) - (FP_{2020})}{(FP_{2022})}, \quad (4)$$

where ( $FP_{2020}$ ) refers to the final parameter metric of Carruba et al. (2020) machine learning applications and ( $FP_{2022}$ ) is linked to the current work metrics.

This problem can be handled by reapplying the Carruba et al. (2020) machine learning algorithms to the updated database of the 21 asteroid families investigated. Table 5 shows the new percentual difference  $R$ .

Figures 1 and 2 show the distribution of the original and retrieved family members in the cases of the Eunomia and Maria families. Those families were chosen based on the metric comparison: 15 Eunomia had the worst value of  $R$  (-0.124), while Maria displayed the best result, with  $R = 0.141$ . In all cases, the algorithms were able to retrieve asteroids in the right orbital regions of the asteroid families, with modest gains for the case of genetic algorithms applied to the Maria family.

The frequency of  $R$  distribution in the 21 studied asteroid families can be seen in Figure 3.

## 4 Conclusion

According to the histogram data (Figure 3), the  $R$  distribution is slightly positively skewed (to the right, where the genetic algorithms perform better) and has more values in the tails than a normal distribution. A small standard deviation shows that the sample values are condensed close to the mean, indicating a homogeneous sample.

The  $R$  fluctuation demonstrates a connection between old machine learning codes and current genetic algorithm implementations. Although there is a minor improvement in

some families' precision and a small drop in others using the genetic method, the overall picture shows that the new application is satisfactory, demonstrating that using the genetic algorithm to solve this problem is safe.

## Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

## Author contributions

ML and VC contributed to the conception and design of the study. VC and ML organized the database. ML and VC performed the statistical analysis. ML and VC wrote the first draft of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

## Funding

We are grateful to the Brazilian National Research Council, CNPq, for supporting the work of ML (grant PIBIC 2021/2698) and VC (grant 304168/2021-1).

## Acknowledgments

This research has made use of the Asteroid Families Portal maintained at the Department of Astronomy/University of Belgrade. We are grateful to Prof. Rita C. Domingos and Safwan Aljbae for useful suggestions and comments.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

<sup>1</sup> In this work, we will refer to an asteroid family with the identification and name of its lowest numbered member, that is, 15 Eunomia, for example. Please note that the correct way to do that would be by using parenthesis around the identification number, that is, (15) Eunomia. For the sake of simplicity, here, we will prefer to use our shortest nomenclature.

## References

- Bendjoya, P., and Zappalà, V. (2002). "Asteroid family identification," in *Asteroids III* (Tucson, AZ, United States: Arizona Univ. Press), 613–618.
- Bremermann, H. (1958). *The evolution of intelligence: The nervous system as a model of its environment*. Washington: University of Washington, Department of Mathematics.
- Carruba, V., Aljbaae, S., Domingos, R. C., Lucchini, A., and Furlaneto, P. (2020). Machine learning classification of new asteroid families members. *Mon. Not. R. Astron. Soc.* 496, 540–549. doi:10.1093/mnras/staa1463
- Carruba, V., Aljbaae, S., and Lucchini, A. (2019). Machine-learning identification of asteroid groups. *Mon. Not. R. Astron. Soc.* 488, 1377–1386. doi:10.1093/mnras/stz1795
- Carruba, V., Domingos, R. C., Nesvorný, D., Roig, F., Huaman, M. E., and Souami, D. (2013). A multidomain approach to asteroid families' identification. *Mon. Not. R. Astron. Soc.* 433, 2075–2096. doi:10.1093/mnras/stt884
- Holland, J. H. (1962). Outline for a logical theory of adaptive systems. *J. ACM* 9, 297–314. doi:10.1145/321127.321128
- Knežević, Z., and Milani, A. (2003). Proper element catalogs and asteroid families. *Astron. Astrophys.* 403, 1165–1173. doi:10.1051/0004-6361:20030475
- Le, T. T., Fu, W., and Moore, J. H. (2020). Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics* 36, 250–256. doi:10.1093/bioinformatics/btz470
- Milani, A., Cellino, A., Knežević, Z., Novaković, B., Spoto, F., and Paolicchi, P. (2014). Asteroid families classification: Exploiting very large datasets. *Icarus* 239, 46–73. doi:10.1016/j.icarus.2014.05.039
- Nesvorný, D., Brož, M., and Carruba, V. (2015). "Identification and dynamical properties of asteroid families," in *Asteroids IV* (Tucson: Univ. of Arizona), 297–321. doi:10.2458/azu\_uapress\_9780816532131-ch016
- Olson, R. S., Bartley, N., Urbanowicz, R. J., and Moore, J. H. (2016a). "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Gecco '16: Proceedings of the genetic and evolutionary computation conference 2016* (New York, NY, USA: ACM), 485–492. doi:10.1145/2908812.2908918
- Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., and Moore, J. H. (2016b). "Applications of evolutionary computation: 19th European conference, EvoApplications 2016, porto, Portugal, march 30 – april 1, 2016, proceedings, Part I," in *Automating biomedical data science through tree-based pipeline optimization* (Springer International Publishing), 123–137. doi:10.1007/978-3-319-31204-0\_9
- Radović, V., Novaković, B., Carruba, V., and Marčeta, D. (2017). An automatic approach to exclude interlopers from asteroid families. *Mon. Not. R. Astron. Soc.* 470, 576–591. doi:10.1093/mnras/stx1273
- Zappalà, V., Bendjoya, P., Cellino, A., Farinella, P., and Froeschlé, C. (1995). Asteroid families: Search of a 12, 487-asteroid sample using two different clustering techniques. *Icarus* 116, 291–314. doi:10.1006/icar.1995.1127