Serb. Astron. J. № 201 (2020), 39 - 48
DOI: https://doi.org/10.2298/SAJ2001039V

UDC 523.44
Original scientific paper

CLASSIFICATION OF ASTEROID FAMILIES WITH ARTIFICIAL NEURAL NETWORKS

D. Vujičić[1], R. Pavlović[2], D. Milošević[1], B. Đorđević[3], S. Ranđić[1] and D. Stojić[1]

[1] *University of Kragujevac, Faculty of Technical Sciences Čačak, Sv. Save 65, 32000 Čačak, Serbia*
E–mail: *dejan.vujicic@ftn.kg.ac.rs*

[2] *Astronomical Observatory, Volgina 7, 11060 Belgrade 38, Serbia*
E–mail: *rpavlovic@aob.rs*

[3] *Institute Mihailo Pupin, Volgina 15, 11060 Belgrade, Serbia*

(Received: August 3, 2020; Accepted: November 27, 2020)

SUMMARY: This paper describes an artificial neural network for classification of asteroids into families. The data used for artificial neural network training and testing were obtained by the Hierarchical Clustering Method (HCM). We have shown that an artificial neural networks can be used as a validation method for the HCM on families with a large number of members.

Key words. Methods: data analysis – Minor planets, asteroids: general

1. INTRODUCTION

The most commonly used method for classification of asteroids into families is the Hierarchical Clustering Method (HCM). In recent years, in its different versions, it has been used almost as the only method for classification of asteroids. In this paper, we propose an artificial neural network (ANN) for classification of asteroid families. It will be shown that this approach can be used as a supplementary validation method with the HCM, however, with certain limitations. These limitations are set by the number of asteroids accessible to the artificial neural network to learn.

This paper is organized as follows. In Section 2 we describe the related work on the HCM for classification of asteroids into families. Section 3 gives the necessary theoretical background of the artificial neural networks, together with the learning methods related to the deep learning algorithms. Deep learning represents a class of machine learning algorithms that uses ANNs with multiple layers of neurons for input data processing (Lecun et al. 2015). The dataset used is described in Section 4, with a detail explanation of the artificial neural network model used for training. Section 5 presents the results and discussion, particularly of the ANN training and testing, and feature importance. The final Section 6. gives the concluding remarks on the topics covered in the paper.

2. RELATED WORK

One of the most used techniques for the classification of asteroids into families is the HCM, presented in (Zappala et al. 1990, 1994). The Wavelet Analysis Method (WAM) is also used (Bendjoya et al. 1991, Baluev and Rodionov 2020), and it is in accordance with the HCM method (Bendjoya and Zappalà 2002, Knežević 2015). The HCM may be applied in domains of proper frequencies, for studying asteroid families interacting with secular resonances (Carruba and Michtchenko 2007, 2009).

Machine learning has been recently used for asteroid classification. In Carruba et al. (2019), the authors used supervised machine learning hierarchical clustering methods for identifying asteroid families in low-number density regions with accuracy above 89.5%. In Carruba et al. (2020), the authors used 13 machine learning classification algorithms for updating of the asteroid family membership. The extremely randomized trees algorithm had the highest precision of 97%.

The HCM method is based on determination of distances between asteroids in a three-dimensional space of proper elements $(a_\mathrm{p}, e_\mathrm{p}, i_\mathrm{p})$, where: a_p is the proper semimajor axis of the asteroid's orbit, e_p is the proper eccentricity of the asteroid's orbit, and i_p is the proper inclination of the asteroid's orbit.

Proper elements are quazi-integrals of the asteroid equations of motion and represent the motion constants in linear approximation. They can be calculated numerically or analytically, by removing the short and long periodic perturbations of the asteroid orbital elements. Proper elements were computed for about 12,000 asteroids in (Milani and Knežević 1994), followed by over 70,000 asteroids in (Knežević et al. 2002). Nowadays, the AstDys service* (Knežević and Milani 2003) includes information about proper elements of more than 650,000 asteroids. This catalog consists of more than 520,000 numbered objects and more than 130,000 unnumbered objects.

The modified HCM method performs the clusterization by identifying the asteroid positions in proper elements space that are within the cutoff distance. When such asteroids are identified, each one of them is used as a center around which asteroids with appropriate distances are identified. The procedure stops once there are no more surrounding asteroids left to belong to a cluster. The value of the cutoff distance determines the number of asteroids being included in a cluster. The greater the distance, the larger number of asteroids is included. Typically, this cutoff distance is between 1 and 200 m/s (Nesvorný et al. 2005).

On the recent classification results, Novaković et al. (2011) performed clusterization of more than 18,000 numbered high-inclination asteroids. In Nesvorný et al. (2015), the authors identified 122 families with more than 100,000 members. Milani et al. (2014) used a catalog with more than 330,000 numbered asteroids, categorizing them into 128 families with around 87,000 members. Milani et al. (2016) followed on and analyzed the dataset with more than 500,000 asteroids. They established 115 families with more than 120,000 members and automated classification updates.

3. ARTIFICIAL NEURAL NETWORKS

ANN consists of neurons and belonging synapses. The neurons in an ANN are organized and function similar to the actual neurons in the human brain. A basic ANN consists of one input and output layer of neurons and the presence of the hidden layers between them improves their precision. This form of ANNs with hidden layers is the foundation of the deep learning paradigm in artificial intelligence areas (Lecun et al. 2015, Goodfellow et al. 2016).

Each neuron in the ANN, except the neurons in the input layer, performs calculation of the weighted sum, Eq. (1). The weights are numerical values assigned to individual synapses and they represent the importance of the corresponding input neuron. In the process of learning, the ANN adjusts these weights, thus increasing or decreasing the importance of certain neurons in the networks:

$$\mathrm{weighted\_sum} = \sum_{i=1}^{m} w_i X_i . \tag{1}$$

In Eq. (1), weights are denoted as w_i, while X signifies the input value to the neuron. This weighted sum is calculated for m inputs. The inputs are independent variables and they represent values supplied to the ANN. They need to be standardized or normalized for proper scaling and ANN functioning. The outputs are dependent variables and they represent the values that the ANN produces. In the case of classification or clusterization, the number of output values is equal to the number of categories.

Fig. 1 shows the ANN with three layers of neurons. It is common to represent the input values as separate input layer neurons, and the output value as separate output layer neurons. This figure also shows one hidden layer consisting of three neurons.

Besides the calculation of the weighted sum, the neurons apply the activation function. This is a function that produces the output from the neuron, with the weighted sum as an input. If we denote the weighted sum as x, then the rectifier activation function, one of the mostly used activation functions (Glorot et al. 2011), is given as in Eq. (2). Other than that, other activation functions commonly used in ANNs are the threshold function (Eq. (3)), sigmoid function (Eq. (4)), and hyperbolic tangent function (Eq. (5)):

$$f(x) = \max(x, 0) , \tag{2}$$

$$f(x) = \left\{ \begin{array}{ll} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{array} \right) , \tag{3}$$

$$f(x) = \frac{1}{1 + e^{-x}} , \tag{4}$$

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} . \tag{5}$$

The symbol $\hat{y}$ in Fig. 1 represents the output or predicted value, which is usually different from the

*https://newton.spacedys.com/astdys

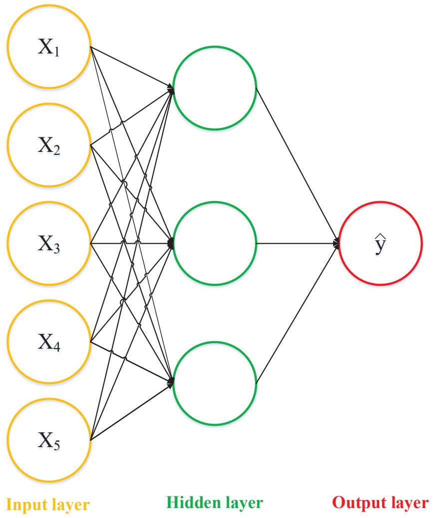

Fig. 1: The ANNs, with five neurons in the input layer, three neurons in the hidden layer, and one neuron in the output layer.

actual output value. After the initial output calculation, the cost function (or loss function) is applied to all weights in the ANN by the algorithm of backpropagation (Rumelhart et al. 1986). The cost function is calculated using the difference between the predicted and actual output values. Some of the cost functions (C) mostly used in ANNs are a squared error (Eq. (6)) and cross-entropy (Eq. (7)). The value of j in these equations indicates the individual training outputs, from the total of n training outputs:

$$C = \frac{1}{2}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2 \,, \tag{6}$$

$$C = -\sum_{j=1}^{n}[y_j \ln \hat{y}_j + (1 - y_j)\ln(1 - \hat{y}_j)] \,. \tag{7}$$

After calculation of the cost function, it is necessary to find its minimum to get the best prediction results possible. The optimization algorithm is applied to the cost function to find its minimum. This algorithm usually begins with finding the gradient of the cost function and is based on that the weights in the ANN are updated. The optimization algorithm operates until the minimum of the cost function is achieved. The usual optimization algorithms used in ANNs are Stochastic gradient descent (Bottou et al. 2018), AdaGrad (Duchi et al. 2011), RMSProp (Zou et al. 2019), and Adam (Kingma and Ba 2014).

The learning process in an ANN starts with initialization of weights with values near zero. Then, the first row of data is forwarded as input variables and then forward propagated through the network. The predicted result is compared with the actual result, the optimization algorithm is applied to the cost function, and with the backpropagation, the weights are updated. These steps are repeated for every observation (or row in the dataset) or after a subset (batch) of observations. The learning process is shown in Fig. 2.

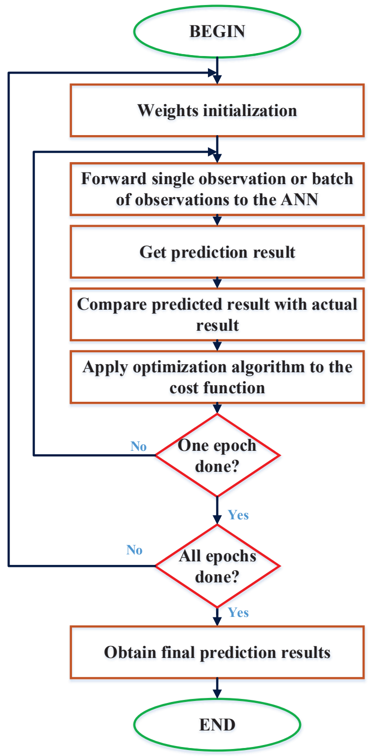

Fig. 2: The learning process in the artificial neural networks.

An epoch is done when all the data from the dataset have passed through the ANN. An epoch in the ANN context represents the completion of the training procedure for all available observations. To make better predictions, this whole process is repeated for an arbitrary number of epochs (Nielsen 2015).

4. DATASET AND MODEL DESCRIPTION

The dataset consists of 124,478 asteroids divided into 110 families and is obtained from the AstDys website[†]. The data for each asteroid consists of its proper semimajor axis (a_p), proper inclination sinus ($\sin(i_\mathrm{p})$), proper eccentricity (e_p), absolute magnitude (H), and the family name. The asteroid families in the dataset can be divided into subsets based on the number of members. The huge families, which contain over 10,001 elements, are Eos (16,038 members), Hertha (15,983 members), and Vesta (10,612 members). Other than these, there are 22 large families with 1,001 to 10,000 members, 39 medium families with 101 to 1,000 members, and 46 small families with up to 100 members.

[†]https://newton.spacedys.com/astdys

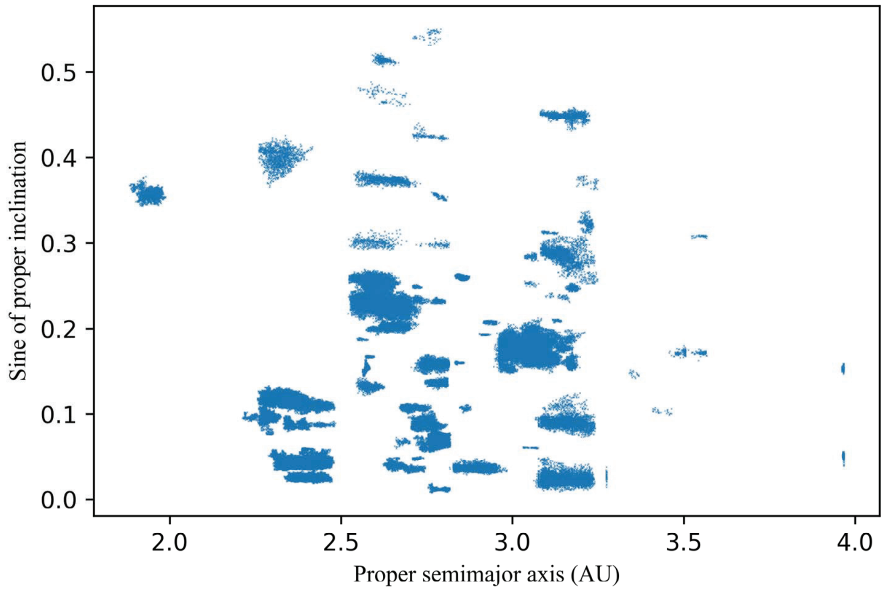

Fig. 3: The distribution of asteroids in the $(a_\mathrm{p}, \sin(i_\mathrm{p}))$ plane.

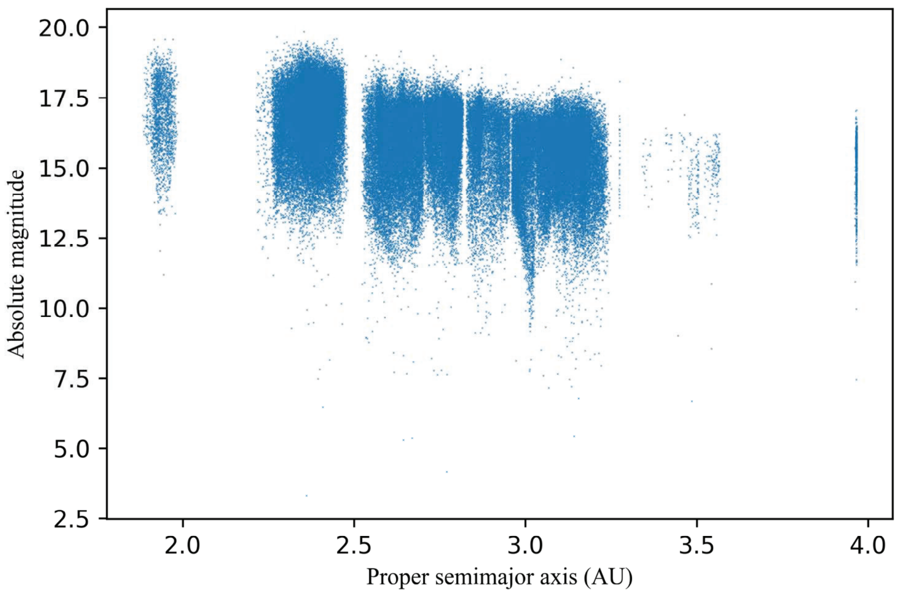

Fig. 4: The distribution of asteroids in (a_p, H) plane.

The distribution of asteroids in the $(a_\mathrm{p}, \sin(i_\mathrm{p}))$ plane is given in Fig. 3. The distribution of data in the (a_p, H) plane is given in Fig. 4. From Fig. 4 it can be seen that few asteroids have $H < 14$, which can be used as a training subset since most asteroids that have $H < 14$ are close to the family center (Carruba et al. 2020). We found that the number of asteroids that satisfy this condition is 12.041, with 6 families not being represented, so we didn't use this method for determining the training subset. Instead, we used the entire dataset with 80% of data used for training, and 20% for testing, making sure that every family is included.

The ANN model is realized in Python programming language, using Keras (Chollet et al. 2018) and Sklearn (Pedregosa et al. 2011) libraries. The input layer consisted of three neurons, for the three input variables (proper semimajor axis, proper eccentricity, and proper inclination sinus). The output layer consisted of 110 neurons, which is equal to the number

BEFORE ENCODING	
	Family
AST1	Vesta
AST2	Eos
AST3	Hertha
AST4	Eos
AST5	Eos
AST6	Hertha

AFTER ENCODING			
	Col1	Col2	Col3
AST1	1	0	0
AST2	0	1	0
AST3	0	0	1
AST4	0	1	0
AST5	0	1	0
AST6	0	0	1

Fig. 5: Binary encoding of six asteroids belonging to three asteroid families.

of families to be classified. A good practice is to use the arithmetic mean of the number of input and output layer neurons for the hidden layer, so we used 56 neurons for the hidden layer.

After loading the data, the independent and dependent variables were set. The input independent variables needed to be scaled and normalized to get better prediction results. Since we got categorical values for the dependent output variable, we needed to encode them with the binary encoder. In this way, instead of a one-column output variable (family name), we got a 110-column output variable. The column equal to one represents the membership of a given asteroid to the corresponding family, while other columns are equal to zero. This process is shown in Fig. 5, in an example of six asteroids belonging to three asteroid families.

The next step was to split the original dataset into training and testing subsets. We opted for 80% of data to belong to the training subtest, and the remaining 20% to the testing subset. This split was randomly performed, but in a way that every family is represented in both subsets. The training subset was used for the ANN training and learning, while the testing subset was used as new data to the ANN, to get the prediction results.

The best results were obtained using the rectifier function as the hidden layer activation function, and the sigmoid function as the output layer activation function. The cross-entropy cost function was used since it provided the best results. As for the optimization algorithm, several algorithms were tested. The test was performed using 20 epochs for the ANN training. The training accuracies achieved on the training subset are given in Table 1. The accuracy is Keras metric for measuring the performance of the ANN model. It is calculated as the percentage of the training subset data that were successfully predicted. Regarding asteroid family classification, the accuracy represents the percentage of correctly predicted family members out of the total number of asteroids used in the training subset. From now on, we will use the term "training accuracy" for the accuracy of the training data, and the term "testing accuracy" for the accuracy of the test data.

Table 1: Training accuracies of the optimization algorithms used.

Optimization algorithm	Training accuracy
RMSProp	95.14 %
AdaGrad	97.09 %
Stochastic Gradient Descent	98.09 %
Adam	99.17 %

As can be seen from Table 1, the Adam optimization algorithm outperformed the other three, so it was used in the next step. Since we opted for the Adam optimization algorithm, we needed to perform k-fold cross-validation. This is a technique in which the training subset is divided into k equal folds. With every cross-validation, the different $k-1$ folds are used for training and the remaining one fold for testing. The optimal number of folds is 10, so we used this number for cross-validation.

5. RESULTS AND DISCUSSION

5.1. Results of artificial neural network training

Cross-validation is useful for validation of the ANN model, particularly regarding the stability of the model with introduction of new data. After performing the 10-fold cross-validation, with 20 training iterations (epochs) in each, the obtained training accuracies are shown in Fig. 6.

The worst obtained training accuracy was 98.82%, and the best was 99.33%. The arithmetic mean of all training accuracies was 99.15% with a standard deviation of 0.17%. These numbers indicate a stable model with a low standard deviation from the mean value of training accuracies (Efron 1983).

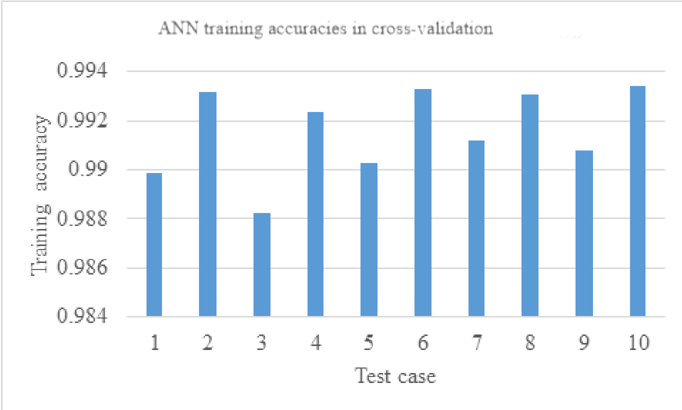

Fig. 6: The training accuracies in 10-fold cross-validation.

5.2. Results of artificial neural network testing

After validating the model with the training data, the next step in ANN validation is the measurement of the accuracy of the test data, which can be thought of as new data for the ANN. The usual technique is the confusion matrix representation of prediction output on the test data. The confusion matrices were formed for each family of asteroids. The rows in the confusion matrix represent the predicted results, while the columns represent the actual results. The confusion matrix consists of a number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions, Fig. 7. True positive is the number of family members that both the HCM and our model identified. True negative is the number of non-family members that both methods identified. False positive is the number of family members that only our model identified as such. False negative is the number of family members not identified as such only by our method.

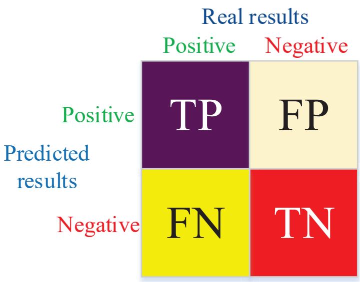

Fig. 7: The confusion matrix.

The testing accuracy of the classification algorithm can be defined as Fawcett (2006):

$$\text{testing\_accuracy} = \frac{TP + TN}{TP + TN + FP + FN}\,. \tag{8}$$

Few other metrics can be used to provide more information on ability of the algorithm to correctly predict asteroid families' members. Those are, among others, precision, recall, merit, and F1-score (Caruana and Niculescu-Mizil 2006). Precision is the measure of the algorithm ability to avoid false data, and is calculated as:

$$\text{precision} = \frac{TP}{TP + FP}\,. \tag{9}$$

Recall, or sensitivity or completeness (Carruba et al. 2020) is the measure of the algorithm ability to retrieve all known family members. It is calculated as:

$$\text{recall} = \frac{TP}{TP + FN}\,. \tag{10}$$

Another metric that is used for validation of precision and recall is F1-score (Fawcett 2006). It is calculated as:

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}\,. \tag{11}$$

Table 2 shows the testing accuracies over some representatives of asteroid families, along with the values of precision, recall, and F1-score. Besides the asteroid family names and confusion matrices-based testing accuracies obtained, the total number of members in all shown families and the number of members used in testing are given. This is done to show that approximately 20% of family members were used as test data, as previously stated.

The testing accuracy of 0% means that none of the asteroids used in testing was correctly classified, while the testing accuracy of 100% represents that all the asteroids used in testing were correctly classified. From Table 2, we can see that for families with a small number of members, the ANN model scored a very large set of accuracies. The largest family with 0% testing accuracy was (31811) 1999 NA41, and of 46 families with less than 100 members, the 0% testing accuracy was achieved in 6 cases. The largest family with a testing accuracy of less than 90% was (5026) Martes.

Precision, recall, and F1-score are in agreement with the accuracy results. However, as can be seen from Table 2, recall is equal to 100% for some families

Table 2: Accuracies, precision, recall, and F1-score obtained from confusion matrices for some asteroid families.

Asteroid family	Total number of members	Number of members used in testing	Precision	Recall	F1–score	Testing accuracy (%)
(69559) 1997 UG5	17	1	0	0	0	0
(260) Huberta	26	2	100	100	100	100
(22805) 1999 RR2	20	3	20	67	31	45
(116763) 2004 EW7	24	4	67	50	57	70
(153) Hilda	18	5	100	100	100	100
(14916) 1993 VV7	17	7	0	0	0	0
(159) Aemilia	62	8	0	0	0	0
(4203) Brucato	41	10	54	70	61	60
(2) Pallas	45	12	67	100	80	72
(43176) 1999 XM196	75	15	32	53	40	45
(6769) Brokoff	58	17	0	0	0	0
(618) Elfriede	97	20	67	100	80	87
(11882) 1990 RA3	87	22	67	100	80	74
(1338) Duponta	133	27	93	100	96	95
(31811) 1999 NA41	144	28	0	0	0	0
(18466) 1995 SU37	257	44	79	100	88	80
(1303) Luthera	232	51	80	100	89	82
(5026) Martes	481	98	73	100	84	86
(110) Lydia	898	175	99	100	99	99
(2076) Levin	1534	315	99	99	98	97
(490) Veritas	2139	425	99	100	99	99
(10) Hygiea	3145	628	99	100	99	99
(24) Themis	5612	1189	100	100	100	100
(158) Koronis	7390	1446	100	100	100	100
(4) Vesta	10612	2038	100	100	100	100
(15) Eunomia	9856	2064	100	100	100	100
(221) Eos	16038	3169	99	99	99	99
(135) Hertha	15983	3233	99	100	100	99

where accuracy and precision are much lower. This would indicate that, for these families, the algorithm was available to retrieve all family members.

The results presented in Table 2 show that the realized ANN can be successfully used as a validation method, supplementing the HCM method. However, it also shows that the number of asteroids in a family significantly affects the ANN prediction. This is something that is expected – the more data the ANN can be trained on, the better testing accuracies it will produce. To summarize the testing accuracies regarding the family size, Table 3 was created. Table 3 shows the average testing accuracy over the subsets of asteroid families, along with a total number of families in each subset, lowest testing accuracy obtained, number of testing accuracies equal to 0%, number of testing accuracies between 1% and 90%, number of testing accuracies between 90% and 99%, and number of testing accuracies equal to 100%.

The data shown in Table 3 clearly shows that the ANN can perform very well, but with asteroid families with a large number of members. The acceptable results are achieved with medium size families, except for Martes. On the other hand, this ANN model cannot be applied to small families, and likewise, cannot be properly used for validation of the HCM results for families with a small number of members.

5.3. Feature importance

To better explain the ANN model, i.e. to determine which inputs have a major contribution to the ANN training accuracy, we used three different combinations of input parameters. In the first combination (C1), the proper semimajor axis was omitted, in the second (C2), the proper eccentricity was omitted, and in the third (C3), the proper inclination sinus was omitted. The ANN model remained the same as in the first part of the training. The results are shown in Table 4.

As can be seen from Table 4, the best training accuracy was obtained for the C2 combination of input parameters. This means that proper eccentricity of the asteroid has the least impact on determination of the belonging family. Also, the worst training accuracy was obtained for the C3 combination of input parameters, meaning that the proper inclination sinus has the greatest importance in determination of the corresponding family.

Table 3: Statistics for subsets of asteroid families.

Family subset	Total number of families	Lowest testing accuracy	Number of 0 % testing accuracies	Number of testing accuracies between 1 % and 90 %	Number of testing accuracies between 90 % and 99 %	Number of 100 % testing accuracies	Average testing accuracy
Small	46	0 %	6	13	2	24	79.61 %
Medium	39	0 %	1	3	4	31	95.31 %
Large	22	97 %	0	0	4	18	99.73 %
Huge	3	99 %	0	0	2	1	99.33 %

Table 4: The training accuracies for the different input parameters.

Input parameters combination	Training accuracy %
$C_1 = (e_\mathrm{p}, \sin(i_\mathrm{p}))$	91.31
$C_2 = (a_\mathrm{p}, \sin(i_\mathrm{p}))$	93.59
$C_3 = (e_\mathrm{p}, a_\mathrm{p})$	83.92

The asteroids inclinations are associated with the z–component of the angular momentum of asteroids and they tend to be less affected by dynamical mobility than asteroid eccentricities. Usually, asteroid families are much more compact in inclination than in eccentricity.

6. CONCLUSION

In this paper, we presented an artificial neural network for classification of asteroids into families. The data used in the learning and testing process were obtained by the HCM method and described in AstDys site[‡]. The proposed ANN model can be used as a validation supplement to the HCM method. However, as we have shown, the ANN model can be used with good results only with families with more than 1,000 members. The satisfactory results were obtained with families with 101 to 1,000 members, while the proposed ANN cannot be used for smaller families since it produces unsatisfactory results.

Given the good results obtained with the asteroid members of large families, this method can be used as a method for quick analysis of the unlabeled asteroids membership status. The artificial neural networks work best if they have a large amount of data to be trained on. Since this is the case with asteroids belonging to large families, our method can be used for quick determination of unlabeled asteroids.

The ANN used in this paper consists of one input, hidden, and output layer. The average training accuracy obtained was 99.15% and the model was confirmed as stable in a 10-fold cross-validation process. However, the testing accuracies vary significantly depending on the size of the family, hence the limitation on the ANN usage in the HCM validation.

Through changing the input parameters, we have shown that proper eccentricity of the asteroid has the least significance in determination of family membership. On the other hand, the proper inclination sinus was shown as having the greatest importance in family classification.

Since we used 80% of all asteroids as a training subset, most of them will be asteroids with $H > 14$, which may be affecting the ANN results. We intended to develop a single ANN that would be capable of classifying asteroids into families. By not choosing only asteroids with $H < 14$ as a training subset, the chaining issue of dynamic families (Milani et al. 2016, Carruba et al. 2020) may be affecting the results and cause overlaps between near groups. For future work, we plan to develop several ANNs for different family subsets (from small to huge) by selecting asteroids with $H < 14$ as a training subset.

Acknowledgements – The authors wish to thank academician Z. Knežević for his valuable support and suggestions. The authors wish to thank Dr. V. Carruba for his valuable comments and suggestions as a peer-reviewer. The work presented in this paper was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, and these results are parts of the Grant No. 451-03-68/2020-14/200132 with University of Kragujevac - Faculty of Technical Sciences Čačak and the Grant No. 451-03-68/2020-14/200002 of Astronomical Observatory.

REFERENCES

Baluev, R. V. and Rodionov, E. I. 2020, CeMDA, 132, 34

Bendjoya, P., Slezak, E., and Froeschle, C. 1991, A&A, 251, 312

Bendjoya, P. and Zappalà, V. 2002, Asteroid Family Identification (Tuscon: University of Arizona Press), 613

Bottou, L., Curtis, F. E., and Nocedal, J. 2018, SIAM Review, 60, 223

Carruba, V. and Michtchenko, T. A. 2007, A&A, 475, 1145

Carruba, V. and Michtchenko, T. A. 2009, A&A, 493, 267

Carruba, V., Aljbaae, S., and Lucchini, A. 2019, MNRAS, 488, 1377

[‡]https://newton.spacedys.com/astdys/index.php?pc=7.0

Carruba, V., Aljbaae, S., Domingos, R. C., Lucchini, A., and Furlaneto, P. 2020, MNRAS, 496, 540

Caruana, R. and Niculescu-Mizil, A. 2006, in Proceedings of the 23rd International Conference on Machine Learning, ICML '06 (New York, NY, USA: Association for Computing Machinery), 161

Chollet, F. et al. 2018, Keras: The Python Deep Learning library

Duchi, J., Hazan, E., and Singer, Y. 2011, Journal of Machine Learning Research, 12, 2121

Efron, B. 1983, Journal of the American Statistical Association, 78, 316

Fawcett, T. 2006, Pattern Recognition Letters, 27, 861

Glorot, X., Bordes, A., and Bengio, Y. 2011, in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 315

Goodfellow, I., Bengio, Y., and Courville, A. 2016, Deep Learning (Cambridge, Mass.: MIT press)

Kingma, D. P. and Ba, J. 2014, arXiv:1412.6980

Knežević, Z., Lemaître, A., and Milani, A. 2002, in Asteroids III, eds. W. F. Bottke Jr., A. Cellino, P. Paolicchi, and R. P. Binzel, The Determination of Asteroid Proper Elements (Tucson: University of Arizona Press), 603

Knežević, Z. and Milani, A. 2003, A&A, 403, 1165

Knežević, Z. 2015, in IAU General Assembly, 29, 2247028

Lecun, Y., Bengio, Y., and Hinton, G. 2015, Natur, 521, 436

Milani, A. and Knežević, Z. 1994, Icar, 107, 219

Milani, A., Cellino, A., Knežević, Z., Novaković, B., Spoto, F., and Paolicchi, P. 2014, Icar, 239, 46

Milani, A., Spoto, F., Knežević, Z., Novaković, B., and Tsirvoulis, G. 2016, in IAU Symposium, Vol. 318, Asteroids: New Observations, New Models, eds. S. R. Chesley, A. Morbidelli, R. Jedicke, and D. Farnocchia, 28

Nesvorný, D., Brož, M., and Carruba, V. 2015, in Asteroids IV, eds. P. Michel, F. E. DeMeo and W. F. Bottke, Identification and Dynamical Properties of Asteroid Families (Tucson: University of Arizona Press), 297

Nesvorný, D., Jedicke, R., Whiteley, R. J., and Ivezić, Ž. 2005, Icar, 173, 132

Nielsen, M. A. 2015, Neural Networks and Deep Learning (San Francisco: Determination Press)

Novaković, B., Cellino, A., and Knežević, Z. 2011, Icar, 216, 69

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. 1986, Natur, 323, 533

Zappala, V., Cellino, A., Farinella, P., and Knežević, Z. 1990, AJ, 100, 2030

Zappala, V., Cellino, A., Farinella, P., and Milani, A. 1994, AJ, 107, 772

Zou, F., Shen, L., Jie, Z., Zhang, W., and Liu, W. 2019, in Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

КЛАСИФИКАЦИЈА АСТЕРОИДА У ФАМИЛИЈЕ ПРИМЕНОМ ВЕШТАЧКИХ НЕУРОНСКИХ МРЕЖА

D. Vujičić[1], R. Pavlović[2], D. Milošević[1], B. Đorđević[3], S. Ranđić[1] and D. Stojić[1]

[1] *University of Kragujevac, Faculty of Technical Sciences Čačak, Sv. Save 65, 32000 Čačak, Serbia*
E–mail: *dejan.vujicic@ftn.kg.ac.rs*

[2] *Astronomical Observatory, Volgina 7, 11060 Belgrade 38, Serbia*
E–mail: *rpavlovic@aob.rs*

[3] *Institute Mihailo Pupin, Volgina 15, 11060 Belgrade, Serbia*

УДК 523.44
Оригинални научни рад

У раду су примењене вештачке неуронске мреже за проверу класификације астероида у фамилије које су добијене применом НСМ метода. Улазни подаци коришћени у неуронским мрежама су подељени у две групе: једна група је коришћена за учење неуронске мреже док је друга употребљена за њено тестирање. У раду је показано да су неуронске мреже поуздане када се примењују на фамилије са великим бројем чланова.