Prediction of Asteroids' Minimum Orbit Intersection Distances Using a Deep Neural Network

Sriniketh Vangaru
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061
sriv04@vt.edu

1. Introduction

1.1. Problem Statement

Asteroids are abundantly common in our solar system as remnants of celestial bodies from a long time past. However, some of them are considered as hazardous to the earth due to potential collisions with the planet, and as such are classified as Potentially Hazardous Asteroids (PHAs). The process of determining if an asteroid is a PHA is a straightforward task, as it simply conditionally checks two attributes of the given asteroid: its Minimum Orbit Intersection Distance (MOID), measured in astronomical units (au) and its Absolute Visual Magnitude (denoted as H), which is a unitless quantity [2]. Between these two attributes, the MOID is more easily interpreted by humans, as it is the shortest distance between two orbit paths in 3-dimensional space [4]. This project aims to predict the MOID of an asteroid with strong accuracy based primarily on other features of the asteroid, such as its diameter, absolute visual magnitude, geometric albedo, eccentricity, etc.

As the MOID of an asteroid is a numerical value that can be located within several different orders of magnitude, this is a regression task. The deep neural network (DNN) model will be given structured data containing different features of an asteroid, and its goal is simply to output the asteroid's MOID, where the "other" orbit for the purposes of the calculation is assumed to be Earth's orbital path. The primary objective is to make the MOID values be as close to the actual MOID values as possible on an unseen dataset to ensure generalization of the artificial neural network model.

1.2. Discussion of Current Practices

The definition of an asteroid's MOID is the 3-dimensional Euclidean distance between any point on the orbit of one satellite and the orbit of another, where both satellites are orbiting the same object. This can be seen in Figure 1. Formally, assuming that e_1 and e_2 are sets of infinitesimally close points representing the two elliptical orbital paths, this is given as:

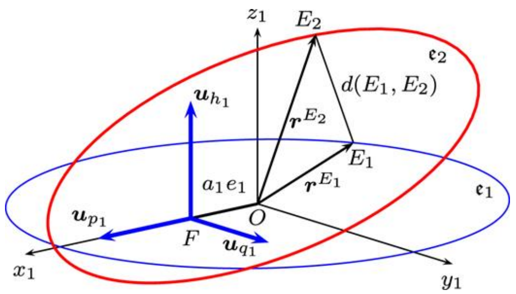


Figure 1. A visualization of the relative geometry of two ellipses [4]. Here, $d(E_1, E_2)$ is an example of an orbit intersection distance.

$$moid(e_1, e_2) = \min_{p_1 \in e_1, p_2 \in e_2} d(p_1, p_2)$$

where $d(x, y) = \sqrt{\sum_{i=1}^{3}(y_i - x_i)^2}$ represents the Euclidean distance between points x and y [4].

However, in order to compute this quantity from the given formula, all information necessary to exactly reproduce the orbit's path is required. As such, there have been alternative methods to compute an asteroid's MOID, such as using trigonometric polynomials [7], applying advanced geometric and algebraic techniques with the Fast Fourier Transform [3], iteratively approximating the MOID by reframing the problem using other variables [10], etc. However, it has not previous been attempted to apply supervised machine learning to this task, which provides the added benefit of being more robust against missing data when approximating the MOID function. Supervised machine learning methods are also generally adept at finding hidden patterns in the data when approximating functions such as the MOID function.

Additionally, the dataset being used for this project has previously been used for similar tasks. Though Hossain and Zabed [6] have used this dataset for a multi-class classification task to predict the orbit class—which essentially describes the path of the orbit—of an asteroid [9], they have also done regression using 7 different types of regression ML models, with a neural network among these models. However, they predicted the diameter, not the MOID, of an asteroid. Interestingly, only the absolute visual magnitude (H) and geometric albedo (a) were used as the inputs to the model. Similarly, Basu [1] also performed regression on asteroid diameter with a similar dataset, again utilizing a multi-layer neural network as one of the models, though more input variables were used.

1.3. Motivation/Significance

A primary use case of this model, if successful, is training making large amounts of MOID predictions for asteroids in an easily scalable manner. As mentioned earlier, a core benefit of implementing a technology such as this in a more impactful setting, such as an astronomical research facility, is that it can pick up patterns from the input data which are not otherwise feasibly interpreted. Specifically, if a model can be trained on particular parameters to predict an asteroid's MOID, then it will demonstrate that data sparsity is not an issue for MOID computation, as other relevant parameters of an asteroid can simply be used. As a result, it will become easier to classify asteroids as PHAs, which is the main purpose of these calculations. In summary, this project presents a different computation of an asteroid's MOID from prior methods because it uses supervised machine learning in its methodology. Additionally, as a byproduct of this, the methods of applying a natural logarithm to the output label and redistributing the data to lessen the impacts of data imbalance and sparsity are also new.

2. Method

2.1. Procedure/Approach

2.1.1 Dataset Processing

The dataset used was obtained from Kaggle, though it was originally drawn from the Small Body Database maintained by NASA's Jet Propulsion Laboratory at the California Institute of Technology [5]. It contains data regarding each of 45 different properties for 958,524 distinct asteroids; however, not every single asteroid had recorded values for each attribute. Some such attributes were the identification label given to each asteroid, its orbit class, whether or not it is a PHA, its MOID, its orbit's eccentricity, its semi-major axis, etc. A considerable amount of data preprocessing was performed, which is described as follows.

The rows of data in which there was no MOID value were removed using Python's `numpy` library, and all other missing values in the dataset were replaced by the median of the corresponding column. Categorical variables were not considered for convenience of data processing. Then, in order to select which input variables would be used to predict the MOID, the correlation coefficient r between each of the attributes in the dataset with the MOID was found, and the top 10 (by absolute value of r) were used, as they would most strongly correlate with the MOID. One attribute that had relatively high correlation with the MOID was H, which was to be expected as they are the two factors that are present in the categorization of an asteroid as a PHA. However, as shown in Figure 2, the two have a logarithmic relationship, so the MOID values were replaced by their natural logarithm as that is a simpler relationship that can even be modeled by a single-layer perceptron. A single-layer perceptron was not used, however, because the correlation coefficient between H and the natural log of MOID was still not strong enough to be used alone (-0.716).

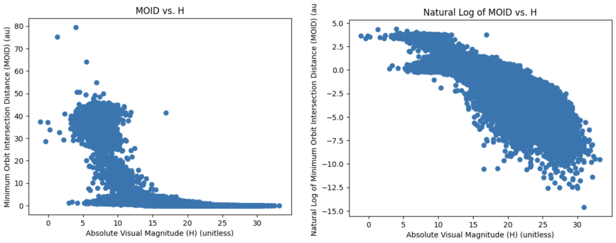
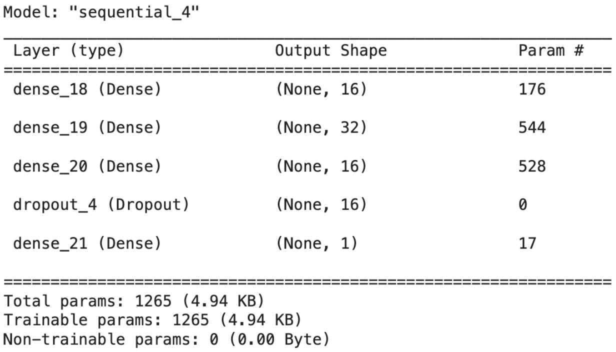


Figure 2. Absolute Visual Magnitude graphed against both Minimum Orbit Intersection Distance and its natural logarithm. The left-side graph has a correlation coefficient of -0.433, and the right-side graph has a correlation coefficient of -0.716.



```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_18 (Dense)            (None, 16)                176

 dense_19 (Dense)            (None, 32)                544

 dense_20 (Dense)            (None, 16)                528

 dropout_4 (Dropout)         (None, 16)                0

 dense_21 (Dense)            (None, 1)                 17

=================================================================
Total params: 1265 (4.94 KB)
Trainable params: 1265 (4.94 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 3. A summary of the DNN model.

Afterward, the Python package `scikit-learn` was used to scale down all the features using the `sklearn.preprocessing.RobustScaler` class. The data rows were then shuffled to avoid any bias through ordering of the training examples. A subset of the data was taken to obtain a relatively more uniform distribution of MOID values, which is further described in §2.1.3. Then, the overall dataset was split into training, cross-validation, and testing datasets.

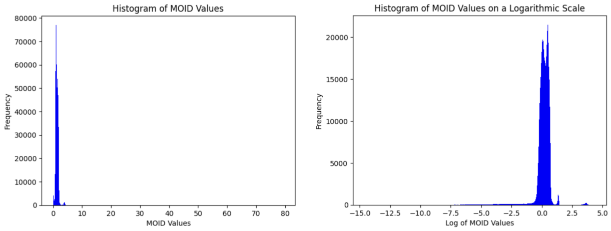


Figure 4. The distribution of the MOID values and their natural logarithms.

2.1.2 Model Training

The model used was a deep neural network (DNN). It used 3 layers, with each layer having between 16 and 32 neurons each, and the only activation function used was the `sigmoid` function. A `Dropout` layer was added after the last `Dense` layer, and the final output layer only had a single neuron with a linear activation function in order to carry out the final regression. All hyperparameters were left as default. A summary of the network itself is given in 3.

2.1.3 Anticipated and Encountered Problems

One problem that was expected to be encountered was **data sparsity**. In particular, there were only 11124 rows of data that satisfied the condition $moid \leq 0.05$, which is approximately 1.185% of the dataset, and this is significant because 0.05 is the threshold used for classifying an asteroid as a PHA. Namely, as the core purpose of obtaining the MOID is to determine how hazardous an asteroid is, a primary obstacle for this project is the imbalance of data in this aspect. It was therefore anticipated that the regression accuracy (as measured by metrics such as Mean Absolute Error (MAE)) would be fairly poor, especially since the scale of MOID values spans several orders of magnitude, as can be seen by the logarithmic scale in 4.

An additional expected issue was **training time**, due to the rather large size of the dataset resulting in more training examples to cover.

To alleviate the first problem, one measure that was taken was to replace the MOID values with the natural log of these values, as mentioned in §2.1.1. Additionally, to counteract both problems, a subset of the data was taken to offset the impact of the largely imbalanced distribution of MOID values. Specifically, after determining the number of rows of data that satisfied the condition $moid \leq 0.05$ (which was 11124), an equivalent number of rows with $moid > 0.05$ were randomly selected. Then, the two subsets of rows were combined, with all the other rows discarded. The impact of these measures upon the results are discussed in §3.1.

2.2. Evaluation Metrics

The evaluation metrics used were standard metrics involved in regression problems. These involved the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (r^2), all of which determine how far the predictions tended to be from the actual label values for the MOID attribute.

Additionally, the resulting numerical outputs were converted into categorical labels. As mentioned in §1.1, for any given asteroid, it is assessed to either be or not be a PHA based on its MOID and its absolute visual magnitude; specifically, it is a PHA if and only if it satisfies $moid \leq 0.05au$ and $H \leq 22.0$ [2]. So, in order to further explore how this project could be of use in the real world, this situation was simulated by converting the MOID values into binary labels based upon whether (positive) or not (negative) they were $\leq 0.05au$. With this binary classification task, the evaluation metrics of accuracy (the number of correct predictions divided by the number of total predictions), precision, recall, and F1-score were used. The latter 3 in particular are of importance, because false negatives or positives could have disastrous consequences in the context of paying attention to particular asteroids that may come near Earth, and these metrics entirely rely upon the datapoints of true/false positives/negatives.

3. Results

3.1. Model Performance

As discussed in sections §2 and §4, multiple methods were used to optimize the performance of the model. The initial performance of the model was relatively poor, with a MAE of 0.484, RMSE of 2.123, and r^2 of -0.00121. The value of r^2 was surprisingly negative, which it should never be, and interpreting the MAE in context, the average prediction was off by approximately 0.5 astronomical units, which is far too large of a distance to be written off as error. In addition, due to the imbalance of the data and consistently same predictions as brought up in §4, there was not a single predicted MOID output that was ≤ 0.05, meaning that for the binary classification task, the precision, recall, and F1-score could not actually be calculated. The accuracy was seemingly good at 0.988, but this is misleading as every single one of the predictions was above the threshold of 0.05 while most of the dataset has MOID values above 0.05. So, for the binary classification task, the inaccurately predicted numerical values were frequently counted as ”correct”.

After modifying all of the MOID values to be on a logarithmic scale, the performance improved slightly, though the errors were still high if interpreted in context, with a MAE of 0.304 and 0 outputted MOID values that were ≤ 0.05.

However, after doing the redistribution of the dataset, the

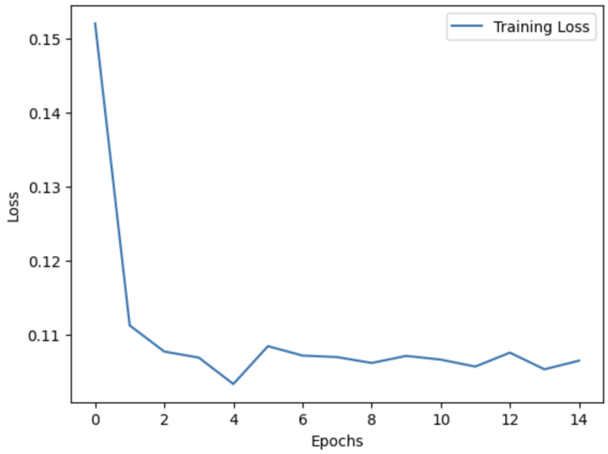


Figure 5. The Mean Squared Error training loss over the training epochs.

performance did improve. The MAE dropped to $0.188au$, with a RMSE of 0.998 and an r^2 of 0.334, showing stronger correlation between the inputs and outputs. Additionally, the outputted values were more distributed across both sides of the threshold of $0.05au$, resulting in an accuracy of 0.979, precision of 0.962, recall of 0.999, and F1-score of 0.97998. The training loss over the training epochs is given in 5. Although the final model was still relatively inaccurate, there was demonstratable improvement as shown by the numbers.

3.2. Availability and Reproducibility

The Python code used in this project is open-source and freely available to use on GitHub [8]. The repository contains the code in a .ipynb file and the trained parameters in text files.

The original data is also available online on Kaggle [5]. The same results obtained are reproducible by way of utilizing a random seed (42) for `numpy` and `tensorflow`. Alternatively, the model parameters can be used to load the model.

4. Discussion

The process of building the model heavily involved the trial-and-error method. Initially, a deeper network was used with 8 layers and the number of neurons (or units) per layer ranging from 16 to 128. However, this did not result in satisfactory performance. The loss went from extremely high in the first epoch to significantly lower (by several orders of magnitude) in the second epoch, and all remaining epochs resulted in the exact same loss as the second. Additionally, during testing, every single test example resulted in the same prediction. It was assumed that this was due to overfitting, so the model was shrunk down; however, although this decreased the training loss slightly, there was still the error of the repeated predictions. However, it was determined that the `relu` activation function was resulting in underfitting from the classic "dying relu" problem where nodes are easily dropped due to too few neurons existing in the network and too many being cancelled out by `relu` automatically setting values to 0, and so the activation function was changed to `sigmoid` for every layer and this alleviated the underfitting that `relu` caused. As a result, the model was able to generalize better to the testing dataset. In general, the model's performance was not successful, though the modifications did improve it by a noticeable amount.

References

[1] Victor Basu. Prediction of asteroid diameter with the help of multi-layer perceptron regressor. *International Journal of Advances in Electronics and Computer Science*, 6(4), 2019.

[2] Paul Chodas. Neo basics: Neo groups. Available at `https://cneos.jpl.nasa.gov/about/neo_groups.html`. Released by NASA's Center for Near Earth Object Studies at the Jet Propulsion Laboratory.

[3] Giovanni F. Gronchi. An algebraic method to compute the critical points of the distance function between two keplerian orbits. *Celestial Mechanics and Dynamical Astronomy*, 93(1):295–329, Sep 2005.

[4] José M Hedo, Manuel Ruíz, and Jesús Peláez. On the Minimum Orbital Intersection Distance Computation: A New Effective Method. *Monthly Notices of the Royal Astronomical Society*, 479(3):3288–3299, 06 2018.

[5] Mir Sakhawat Hossain. Asteroid dataset, 2023. Available at `https://www.kaggle.com/datasets/sakhawat18/asteroid-dataset/data`.

[6] Mir Sakhawat Hossain and Md. Akib Zabed. Machine learning approaches for classification and diameter prediction of asteroids. In Mohiuddin Ahmad, Mohammad Shorif Uddin, and Yeong Min Jang, editors, *Proceedings of International Conference on Information and Communication Technology for Development*, pages 43–55, Singapore, 2023. Springer Nature Singapore.

[7] Konstantin V. Kholshevnikov and Nikolay N. Vassiliev. On the distance function between two keplerian elliptic orbits. *Celestial Mechanics and Dynamical Astronomy*, 75(2):75–83, 1999.

[8] Sriniketh Vangaru. Asteroid minimum orbit intersection distance (moid) regression with a deep neural network (dnn), 2024. Available at `https://github.com/snkv04/asteroid_moid_regression_dnn/tree/main`.

[9] Elizabeth Warner and Ludmilla Kolokolova. Object classifications. Available at `https://pdssbn.astro.umd.edu/data_other/objclass.shtml.`, 2024. Released by NASA's Planetary Data System.

[10] Tomek Wisniowski and H. Rickman. Fast geometric method for calculating accurate minimum orbit intersection distances (moids). *Acta Astronomica*, 63:293–307, 06 2013.