

Cislunar Periodic Orbit Family Classification from Astrometric and Photometric Observations Using Machine Learning¹

**Greg Martin¹, Charles J. Wetterer¹, Jenna Lau¹, Jeremy Case¹,
Nathan Toner¹, C. Channing Chow¹ and Phan Dao²**

¹Pacific Defense Solutions – a Centauri Company¹

²Air Force Research Laboratory

CONFERENCE PAPER

As a means of aiding initial orbit determination for objects in cislunar space, we train various machine-learning algorithms to predict membership in one of 31 elemental periodic orbit families (Doedel, et al., 2007) (e.g. HN2 for northern halo orbit about the $\mathcal{L}2$ libration point) using observational features, including uncertainties, generated from the synodic positions and velocities. Additionally, various methods are investigated to allow for “null” membership where the observation is assessed to not belong to any of the periodic orbit families. The machine-learning algorithms investigated include k-nearest neighbor, support vector machines, decision tree ensembles, random forests, and deep neural networks. The features used include combinations of right ascension, declination, range, their associated rates, and visual magnitude. Individual and integrated performance results are reported and, as expected, vary depending on the features used and uncertainty in the features. Performance accuracy of up to 98% is possible.

1. INTRODUCTION

Orbits in cislunar space cannot be approximated as simple ellipses (i.e. Keplerian) due to the Moon’s gravity playing a more prominent role in the dynamics of the region. As such, applying traditional near-Earth detection, tracking, and initial orbit determination (IOD) routines is expected to be more difficult at best and completely fail at worst. Focusing on the latter, since IOD is a precursor to the formation of a critical component to space domain awareness (SDA)—a space object catalog—the methods presented in this paper aim to bolster (e.g. increase timeliness and accuracy of solutions) IOD techniques by helping to constrain the admissible regions to more probable subsets of cislunar space.

Traditionally, geometric quantities (e.g. semi-major axis, eccentricity, inclination, argument of periapsis, right ascension of the ascending node, mean anomaly) are used to parametrize orbits into classes/regimes/zones (e.g. LEO, MEO, GEO, GTO, HEO). This parametrization only works because of the underlying assumption that orbits are ellipses². Since cislunar orbits are not

¹ Approved for release; distribution is unlimited. Public Affairs release approval # AFMC-2020-0359.

² A circle is the degenerate case of an ellipse.

ellipses, in general, these parameters do not universally translate into a non-Keplerian domain. Categorizing orbits must now rely on other dynamical systems features such as energy (e.g. Jacobi constant), stability indices (e.g. Lyapunov exponent, Floquet multiplier), and solution types (e.g. periodic orbits, quasi-periodic orbits, invariant manifolds, homo-/hetero-clinic connections).

As we expand into cislunar space, we first need to test the art of the possible. Under optimal circumstances with the simplest 3-body orbits, can we use machine learning to determine membership of orbits to families by investigating their measurement space (i.e., classify)?

Starting with the elemental periodic orbits (Doedel, et al., 2007) of the circular restricted 3-body problem (CR3BP), we form a parametrization space over 31 distinct families (Chow, et al., 2020). In this paper we demonstrate the ability of machine learning to classify a cislunar object into one of these cislunar periodic orbit families using a series of observation features (e.g. a track) down to a single observation.

2. DESCRIPTION OF DATA SET

The data set of the cislunar periodic orbit families is described in detail in (Chow, et al., 2020). Each of the 31 continuous manifolds is discretized into 512 orbits. Each orbit records an ephemeris composed of 1024 points separated equally in time over the span of the orbit's period (the non-zero timeframe over which the initial conditions repeat in the synodic frame). Thus, 524,288 points define each manifold or family. For any given observation time or vector of times, a sample of features can be extracted using these points or a subset of these points. Figure 1 illustrates a single such case (orbit 100 of 512 in the HN2 family) in synodic coordinates (Earth at $[-\mu, 0, 0]$ and Moon at $[1-\mu, 0, 0]$), where μ is the mass ratio of the CR3BP. The period of this particular orbit is 14.46 days.

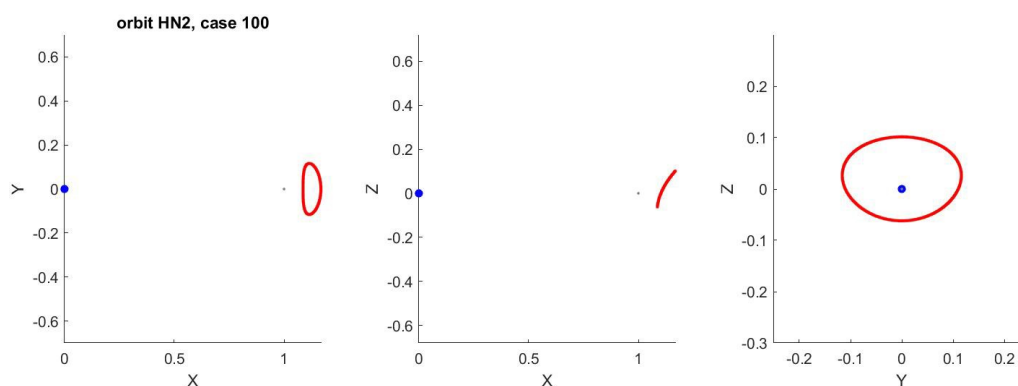


Figure 1. Case 100 periodic orbit in HN2 family in synodic coordinates

For the analysis in this paper, the state of object i at time t observed from site S is described by feature vector $\mathbf{f}_i^S(t)$. Typically, the features used were right ascension (RA), declination (DEC), and their rates. Feature vectors (i.e. observations) are collected over time into a sample $\mathbf{x}_i^S = [\mathbf{f}_i^S(t_1), \mathbf{f}_i^S(t_2), \dots, \mathbf{f}_i^S(t_k)]^T, \forall j = 1, \dots, k$, where k is the number of observations in a sample. Noise is added to these samples as $\tilde{\mathbf{x}}_i^S = \mathbf{x}_i^S + \mathbf{w}$, where \mathbf{w} is zero-mean Gaussian noise with a standard deviation of 2" in RA and DEC, and 0.00417"/s in their rates.

Here, we construct three datasets. In the first set, a single observation composes each sample, i.e. $k = 1$. In the second set, $k = 16$ observations made over a 2-hour period compose each sample. In the third set, $k = 16$ observations made over an 8-day period compose each sample. For all three datasets, the collection starts at JD = 2458703.458 (JD = 2458703.5 = 2019 Aug 08 UT 00:00 on about first quarter moon).

Two sensor sites are simulated, both with the Sun below the local horizon during observations: Azores ($\lambda=37.8^\circ, \ell=-25.5^\circ$) and Maldives ($\lambda=3.7^\circ, \ell=73.0^\circ$). The Moon is above the horizon at the Azores site and below the horizon at the Maldives site. Between the two of them, these sites cover most of the visible cislunar points on that day.

For some of the algorithms, the evaluation samples are extracted from the training samples in the standard paradigm of 80% training, 10% evaluation, and 10% validation. For the other algorithms, the full set of training samples is made available for training and a separate evaluation set is created. This additional evaluation set contains 1,000 evaluation samples at each sensor site for each family and was generated by selecting random points on each family manifold. These evaluation samples are interpolated to be between the cases and the position in orbit of the original set of 524,288 points per family used in generating the training samples. Of the resulting 31,000 evaluation samples, only about half are visible from each sensor site, and only a fraction of these were used in the final performance evaluation.

Finally, a common set of 2,000 "null" evaluation samples at each sensor site was generated which do not belong in any of the cislunar families. This was achieved by selecting a random starting point from the original evaluation set, adding a 20" Gaussian noise to the starting RA and DEC, adding offsets to the associated rates ($(1+0.5 \times \text{randn})$ in RA and $(-1+0.1 \times \text{randn})$ in DEC), and applying a randomly chosen rate acceleration as observed in one of the other evaluation samples. These so-called "null" family samples were used to test each algorithm's ability to correctly classify evaluation samples that do not belong to any cislunar periodic family.

3. SCENARIO DESCRIPTION

Figure 2 outlines the concept of operations applied in this study. In practice, a particular set of observations would be assembled into an evaluation sample composed of a set of features observed at a particular time (for a single observation) or vector of times (for multiple observations). The complete set of periodic orbit family manifold points as described in the previous section are then used to generate the required training samples with the same features as the evaluation structure. These training samples are used to train the machine-learning algorithm(s). Finally, the trained machine-learning algorithm(s) classify the evaluation sample

into a predicted cislunar periodic family, or into the “null” family if applicable. The predicted family is compared to the truth to evaluate performance.

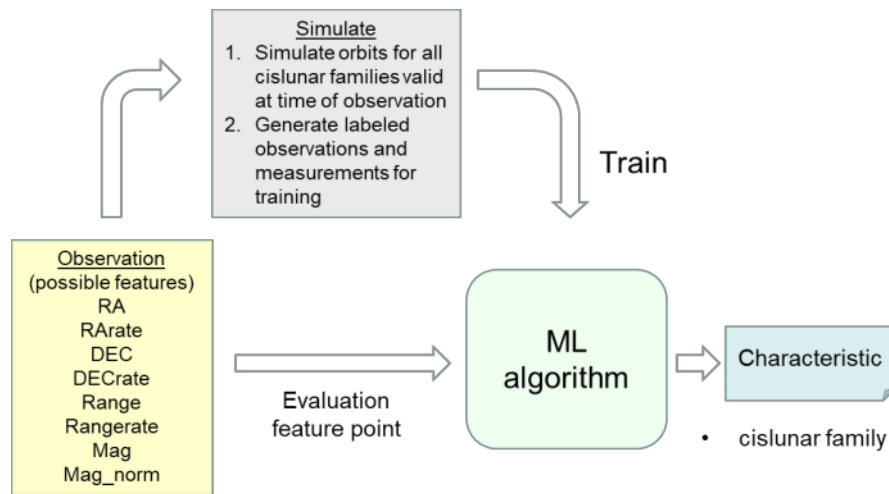


Figure 2. Concept of Operations

For this analysis, the evaluation samples (whether garnered from the training samples, the additional evaluation samples, or the “null” evaluation samples) were generated for the same observation times and sensor sites. For some of the machine-learning algorithms (KNN, DT, and SVM), the algorithm had available the full set of training samples and was trained on a subset of these for each of the “additional” and “null” evaluation samples separately. For the other algorithms (DNN and RFC), the algorithm was trained on 80% of the training samples and the evaluation samples (10% of the training samples and the “null” evaluation samples) were applied to the same trained algorithm.

4. MACHINE-LEARNING ALGORITHMS

Table 1 lists the machine-learning algorithms applied to the classification problem defined in this paper. The rest of this section highlights details about each.

Table 1. List of machine-learning algorithms.

Name	Language	Short Description
KNN	Matlab	K-nearest neighbor. Uses Matlab’s fitcknn and predict functions.
DT	Matlab	Binary decision tree “orchard”. Uses Matlab’s fitctree and predict functions.

SVM	Matlab	Support Vector Machine (both one-class and two-class). Uses Matlab's fitcsvm and predict functions.
RFC	Python	Random Forest Classifier (scikit-learn).
DNN	Python, TensorFlow/Keras	Deep Neural Network (Conv1D) based on a residual network architecture
DNN-AE	Python, TensorFlow/Keras	Deep neural network autoencoder

KNN

The k-nearest neighbors (KNN) classifier uses Matlab's built-in KNN classifier function *fitcknn* to construct the model and the predict function to assess each evaluation sample. This KNN algorithm takes as input a test sample from the evaluation set of samples and compares it with all samples from the training set within a neighborhood to determine to which manifold it is most similar. The search neighborhood for these training samples is set by a hyperparameter and was restricted in RA and DEC to those samples within 0.25° of the evaluation sample being assessed to reduce computation time. A minimum of 1,000 training samples was required, with the RA/DEC window increased if necessary, until the minimum threshold was reached. In this analysis, the *fitcknn* algorithm used the following settings beyond the default: NumNeighbors = 5, DistanceWeight = 'inverse', Standardize = 'on'.

When assessing for only the best cislunar family membership, the orbit family with highest confidence is taken as the output of the KNN. The KNN also provides a score for each evaluation sample tested. When the "null" family was being included, a threshold was set, and the prediction score was used to evaluate if the point was "null" (below the threshold) or in one of the cislunar periodic families (above the threshold). For this analysis, the threshold was set to obtain the highest combined accuracy from both sites.

DT

The decision tree (DT) classifier uses Matlab's built-in DT classifier function *fitctree* to construct multiple models and predict the orbit family of each evaluation sample through the orchard of decision trees. As with KNN, training samples were restricted about each evaluation sample to a minimum of 1,000 and now a maximum of 5,000. *fitctree* was employed as a binary classifier, and so 30 different DTs were needed to classify into the 31 cislunar periodic families. In this implementation, the families were divided in a particular way, and so the resulting binary DT models form an "orchard" as defined in **Figure 3** (the location of each DT shown as a unique red number).

more of the one-class scores was above the threshold, however, the prediction into a cislunar family was applied as before. The threshold was set to obtain the highest combined accuracy from both sites.

RFC

The random forest classifier (RFC) makes use of the scikit-learn RandomForestClassifier utility. An RFC is a meta estimator that fits a number of DTs on various sub-samples of the dataset and uses averaging, or majority voting, to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement.

The input feature set consisted of 16 samples of features: RA, DEC, and their rates. The data sets were separated into train, validation and test data. The entire dataset was randomly split at the orbit level into the three groups with: 80% of the orbits used in training; 10% for validation (after every epoch) and 10% for test (used after hyper-parameter model tuning). During training, the data was randomly shuffled (after every epoch).

Datasets 2 and 3 feature observation as a function of time: 16 observations over a 2-hour observation period and; 16 observations over an 8-day observation period.

For dataset 1, the orbit data was collected as observed as the same time. In order to train and evaluate the data of dataset 1 in the same manner as for datasets 2 and 3, 16-contiguous-sample sequences were constructed for each orbit such that there were no outages.

There are four cases when an outage can occur: 1) object below the horizon; 2) sun above the horizon; 3) object within the moon exclusion zone; 4) no magnitude (NaN) – no reflected light from the object towards the observer.

By this means, there was a substantial reduction in the useable size of the datasets. For the Azores observation site, the usable data was 55%. For Maldives, the usable data was 32%

For dataset 2, the 2-hour observation period, the usable data was 44% for Azores and 27% for Maldives observation sites.

For dataset 3, the 8-day observation period, there were no sequences of 16 observations in which there were not any outages for both Azores and Maldives observation sites. Thus, the entire data set was used was used for dataset 3.

DNN

A custom-built deep neural network (DNN) classifier based on a residual network (ResNet) architecture (He, et al., 2015) was designed and trained to perform the classification task. This DNN is composed of several ResNet-like blocks, each comprising multiple instances of 1D convolution followed by batch normalization followed by leaky ReLU activation. After the final ResNet block, a dropout layer and two dense layers form the network prediction over the 31 cislunar orbit classes. The DNN uses the same dataset as the RFC.

The DNN architecture is shown at a high level in Figure 4. It comprises an encoder E , followed by a fully connected classifier layer FC . The network was trained to minimize categorical cross-

entropy (1) using an Adam optimizer with an initial learning rate of 1e-4. Note that in (1), M is the total number of classes in the dataset (31 in our case), $FC_c(E(\mathbf{x})) = y_c = 1$ if the sample is evaluated to fall into class c and $y_c = 0$ otherwise, and p_c is the probability that a sample will fall into class c .

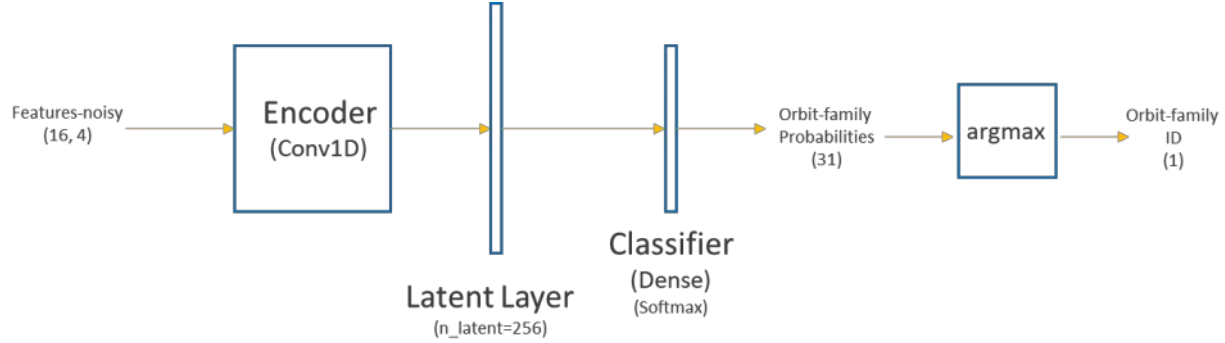


Figure 4. DNN classifier high-level architecture

$$L_{CE} = - \sum_{c=1}^M y_c \log(p_c) \quad (1)$$

Additionally, for some tests a secondary denoising autoencoder was trained. This architecture adds a decoder D that learns to approximate ground truth inputs from the encoded inputs, i.e. $\hat{\mathbf{x}} = D(E(\mathbf{x}))$ such that $\hat{\mathbf{x}} \approx \mathbf{x}$. The decoder comprises an LSTM that takes as input

variables from the last ResNet block and tries to reproduce ground truth samples \mathbf{x}_i from noisy input samples \mathbf{x}_i . The denoised samples $\hat{\mathbf{x}}_i$ could be used in IOD.

This DNN autoencoder (DNN-AE) architecture is shown at a high level in Figure 5. For these trials, a second mean-squared error loss between reproduced and ground truth samples (2) was minimized in addition to categorical cross-entropy. Note that in (2), summation is taken across dimensions to produce a single value for N input samples. The overall loss is a weighted combination of the classifier and decoder losses (3).

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \quad (2)$$

$$L = L_{CE} + \lambda L_{MSE} \quad (3)$$

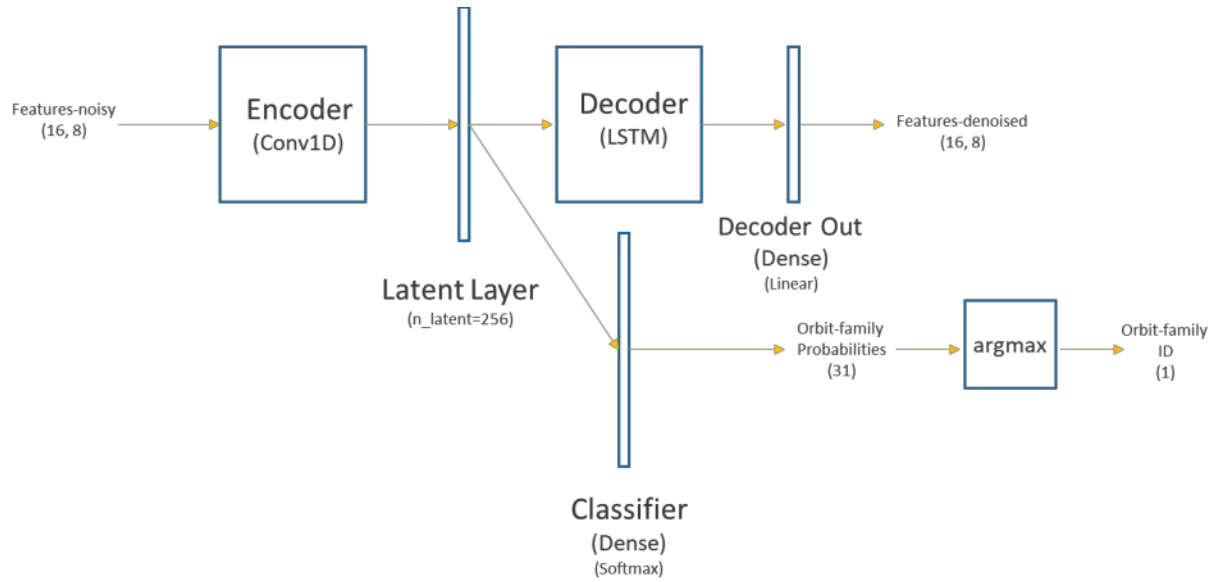


Figure 5. DNN classifier with denoising autoencoder high level architecture

5. RESULTS (KNN, DT, SVM)

Fourteen separate analyses were conducted using the KNN, DT, and SVM algorithms to examine the dependence of the resulting performance on various changes in the observations or training samples. In all analyses, the feature set consisted of the exact values for RA, DEC, and the associated rates (RARate, DECrate). The baseline analysis included all the training data and all the observations per feature (16 million samples with 16 observations each over 2 hours (#1 = 16M16obs/2)). The next four analyses explore the influence of the number of observations per sample by halving the number of observations down to a single observation (#2 = 16M8obs/2, #3 = 16M4obs/2, #4 = 16M2obs/2, #5 = 16M1obs). The next four analyses along with #5 explored the influence of the total number of training samples by halving the number down to one million training samples (#6 = 8M1obs, #7 = 4M1obs, #8 = 2M1obs, #9 = 1M1obs). The next two analyses along with #4 explored the influence of the time span between the first and last observation by selecting the first observation and another in each sample (#10 = 16M2obs/1, #11 = 16M2obs/0.13). The next two analyses along with #5 explore the influence of uncertainty by adding uncertainty to each feature (as described previously) and then double the uncertainty to the exact values in the training and evaluation samples (#12 = 16M1obs w/error, #13 = 16M1obs w/2×error). The final analysis includes the “null” set evaluation samples and allows the algorithms to predict “null” to also compare to #5 (#14 = 16M1obs w/null).

Table 2 and Table 3 list the fourteen separate accuracies for each algorithm from each analysis for the Azores and Maldives sites separately. In both tables, every 5th evaluation sample visible from the particular sensor site (above horizon and outside Moon’s exclusion zone of 1° from Moon’s center) was used (corresponding to 2708 and 1666 for Azores (Table 2) and Maldives (Table 3) respectively for analyses #1-4, #10 and #11 which required the first and last observation to be observable, and 3767 and 2199 respectively for analyses #5-9 and #12-14 which required just the first observation to be observable). In the final analysis, every other

“null” evaluation sample (for a total of 1000 for each sensor site) was also used and added to these totals in computing the algorithm’s performance (for a total of 5767 and 4199 for Azores and Maldives respectively).

Table 2. Accuracy (RA, DEC, RArate, DECrate features) - Azores

Name	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
KNN	0.855	0.869	0.880	0.902	0.853	0.838	0.807	0.746	0.687	0.873	0.847	0.842	0.840	0.675
DT	0.921	0.919	0.925	0.935	0.914	0.884	0.845	0.779	0.713	0.924	0.923	0.892	0.880	0.722
SVM	0.860	0.883	0.903	0.924	0.881	0.854	0.824	0.764	0.697	0.903	0.874	0.865	0.859	0.826

Table 3. Accuracy (RA, DEC, RArate, DECrate features) - Maldives

Name	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
KNN	0.711	0.722	0.744	0.774	0.748	0.726	0.682	0.625	0.560	0.733	0.731	0.740	0.726	0.525
DT	0.836	0.839	0.843	0.854	0.865	0.833	0.786	0.744	0.662	0.849	0.840	0.834	0.805	0.595
SVM	0.720	0.742	0.760	0.799	0.785	0.742	0.699	0.635	0.577	0.770	0.765	0.768	0.752	0.698

The accuracy as a function of the number of observations in each feature while keeping the time span the same (using analyses #1-#5) is shown in **Figure 6**. While going from one observation to two observations with a 2-hour time span slightly increases performance for all algorithms, increasing the number of observations within that 2-hour time span slightly degrades performance.

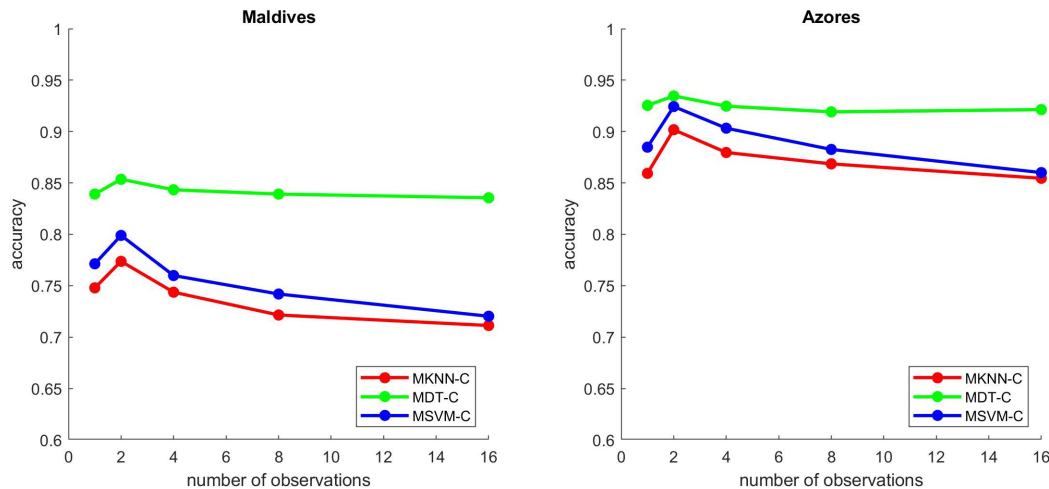


Figure 6. Accuracy with exact data as function of observation length

The accuracy as a function of the total number of training samples with a single observation (using analyses #5-#9) is shown in **Figure 7**. Decreasing the density of training samples

significantly degrades the algorithm performance. The difference in the Azores and Maldives sites highlight the fact that some families are more susceptible to the reduction of training samples than others. For Maldives, where the Moon is below the horizon, the families observed are those with much larger manifolds and thus a lower training sample density. It appears desirable to have a high density in at least one dimension (in this case, along orbit) to best define each manifold. The crosses in **Figure 7** are for an additional analysis where the training samples were randomly divided into 80% to be used for training ($0.8 \times 16\text{M} = 12.8\text{M}$), and 20% to be used for evaluation (of those, 2500 were randomly selected to evaluate for each site). In general, the algorithms perform better than expected. We suspect this is because the evaluation samples for this analysis all fall in the same orbits as the training samples and thus are more likely to be closer together while the “additional” evaluation samples from before were purposely computed to be between the training sample orbits.

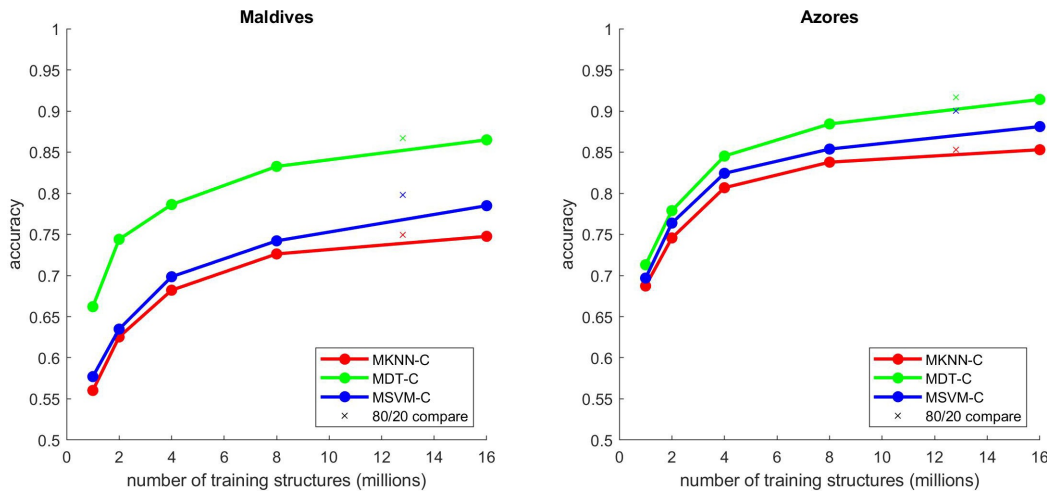


Figure 7. Accuracy with exact data as function of total number of training samples

Figure 8 plots the accuracy as a function of the time span between observations (using analyses #4, #10, #11 and compared to the single observation case #5 with a time span equal to zero). Increasing the time span of feature vectors while keeping the number of observations the same also generally improves algorithm performance, but again not significantly. It is interesting to note that going from one observation (zero time span) to two observations with a 0.13-hour time span slightly decreases performance for all algorithms. This is consistent with **Figure 6** where increasing the number of observations with a similar time span between observations degrades performance.

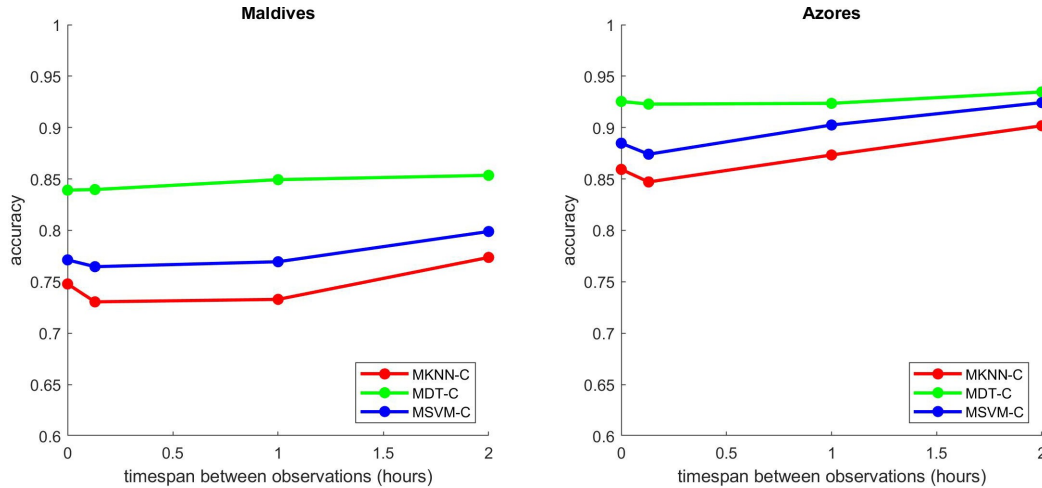


Figure 8. Accuracy with exact data as function of time span between two observations.

Figure 9 plots accuracy as a function of uncertainty (using analyses #5, #12, #13). Noisy data slightly degrades the algorithm performance, affecting some algorithms more than others. This was to be expected.

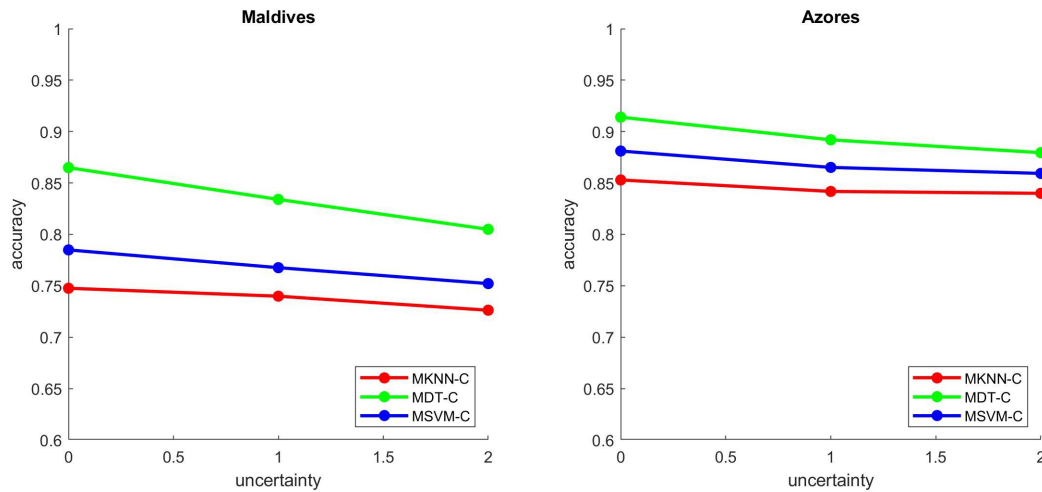


Figure 9. Accuracy with degree of uncertainty added to training and evaluation samples

Finally, adding the “null” set as a possible outcome also significantly degrades algorithm performance (#5 vs. #14). In this case, the thresholds for the scores as calculated by the Matlab predict function (KNN and DT) or the one-class minimum score (SVM) below which the samples were classified as “null” were adjusted to obtain the highest combined accuracy for both sites and the resulting individual accuracies are listed in the tables. The Receiver Operating Characteristic (ROC) curve for each is displayed in **Figure 10** with the point of highest combined accuracy specified.

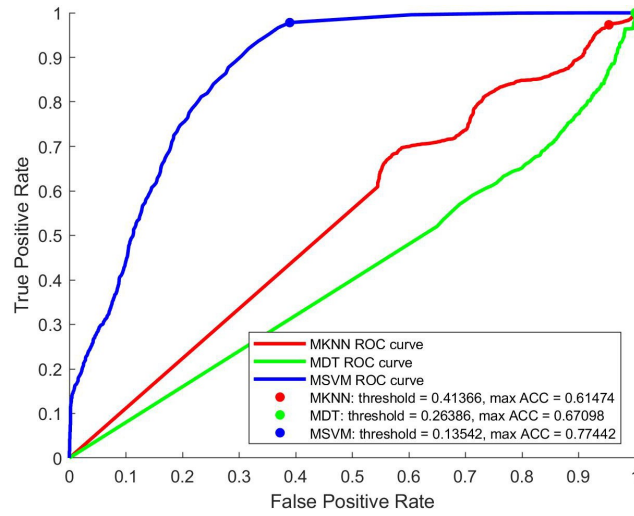


Figure 10. Receiver Operating Characteristic (ROC) curves for algorithms and analysis #14

The ROC curves were constructed where “positive” is in any family (not necessarily a correct prediction) and “negative” is null. Accuracy calculations, however, require correct family or null characterization. The DT algorithm has a ROC curve whose line goes below the diagonal indicating the score calculated is not an adequate discriminator for determining “null” characterization and needs to be replaced. Adjustments to how the individual DT scores were combined into a composite score did invert and improve the appearance of the ROC curve, but did not significantly improve the resulting overall accuracy. The KNN algorithm has a ROC curve just above diagonal indicating the score as calculated is just a marginal discriminator and needs to be improved. Future work to use a score calculated from the scaled distance to the nearest neighbor instead of the function’s raw score shows great promise in improving this algorithm’s null discrimination performance. Only the SVM algorithm has an expected ROC curve with a reasonable accuracy vs. threshold curve and peak at threshold = 0.13542.

6. RESULTS (DNN, RFC)

The overall accuracy versus epoch number for the DNN trained on dataset 1 (all samples observed at the same time), Azores observation site, is shown in **Figure 11** and **Table 4**.

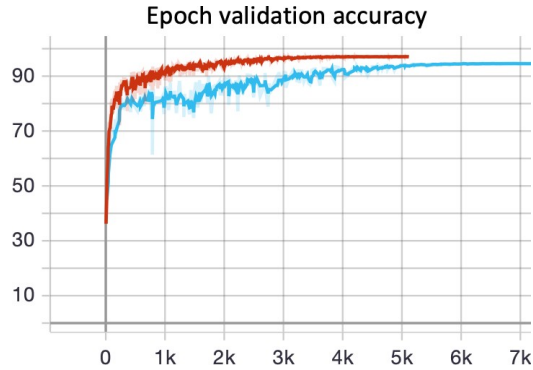


Figure 11. Overall validation accuracy versus epoch number for the DNN model trained on dataset 1 (Azores). Dark red: pristine (no noise) data; Light blue: noisy data.

Table 4. Overall validation accuracy for the DNN model trained on dataset 1 (16 observations at the same time). Observation sites: Azores and; Maldives. Data integrity: no noise and; noise added.

site\noise	No Noise	Noise Added
Azores	0.98	0.97
Maldives	0.98	0.97

The accuracy is seen to be quite good: 98% for the noise-free data and 97% for the noisy data. Similar results were obtained for the Maldives observation site.

The performance results of training and evaluating the DNN with dataset 2 (16 observations over 2 hours) are shown in Table 5.

Table 5. Overall validation accuracy for the DNN model trained on dataset 2 (16 observations over a 2-hour period). Observation sites: Azores and; Maldives. Data integrity: no noise and; noise added.

site\noise	No Noise	Noise Added
Azores	0.88	0.92
Maldives	0.79	0.79

The DNN performance with dataset 2 was not as good as with dataset 1. The reason is believed to be an insufficient observation period to obtain reasonably good estimates of the orbit family. Interesting to note in Table 5 is that the performance improves for the Azores site when noise is added.

The performance results of training and evaluating the DNN with dataset 3 (16 observations over 8 days) are shown in Table 6.

Table 6. Overall validation accuracy for the DNN model trained on dataset 3 (16 observations over an 8-day period). Observation sites: Azores and; Maldives. Data integrity: no noise and; noise added.

site\noise	No Noise	Noise Added
Azores	0.96	0.96
Maldives	0.96	0.97

The overall performance of the DNN trained and evaluated with dataset 3 (16 observations over an 8-day period) is quite comparable to that of the DNN trained with dataset 1 (16 observations all at the same time). Note that we evaluated intermediate observation periods between the 2-hour and 8-day observation period with poor performance compared to the 8-day observation period. We attribute this to not having enough of an orbit segment to suitably discriminate among the orbit families.

For the 8-day observation period (dataset 3), there was not much difference in observed performance between the Azores and Maldives observations sites, probably because all the observations were used, and outages were ignored. An improved sampling scheme, where the observations were adjusted in time where there were no outages, might yield a different result.

The DNN trained model (trained with dataset 3) was analyzed using the “evaluation” and “null” datasets. Recall that the “evaluation” dataset contains 31,000 observations (1000 observations for each of the 31 orbit families). Uncertainty was added to each observation, as described previously. The “null” dataset contains 2,000 observations total where each observation was perturbed to fall outside the manifold of any of the cislunar orbit families. **Figure 12** shows the box plots of both correct and incorrect inferences obtained with the DNN trained model using the “evaluation” dataset broken out by orbit category.

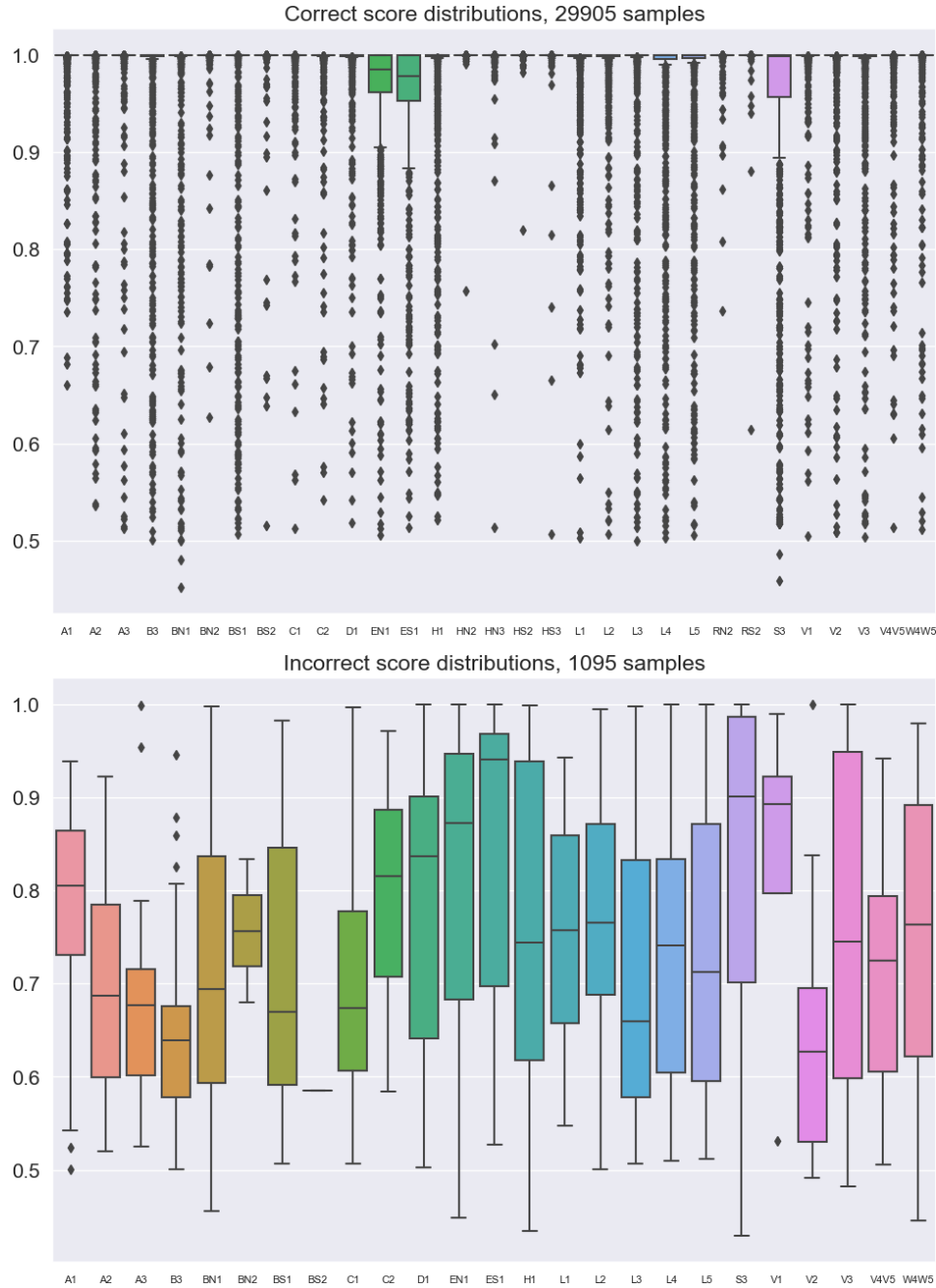


Figure 12. Distribution of confidence scores for the DNN trained on dataset 3 (16 samples observed over an 8-day observation period, uncertainty added) and evaluated using the “evaluation” dataset (31,000 examples, 1000 examples/orbit family). Top graph: Correct orbit family prediction; Bottom Graph: Incorrect orbit family prediction.

As seen from the top graph, of the 31,000 evaluations, the correct orbit family was predicted for 29,905, resulting in an overall accuracy of 0.97. Note the wide range of confidence scores, varying from a very high confidence (1.0) to a marginal result (0.4). The bottom graph shows the distribution of the incorrect scores which also range from 1.0 to about 0.5; however, the average values are considerably less than obtained with the correct score prediction. Note that the bottom

of **Figure 12** represents only about 1000 evaluations in which the wrong category was chosen, and so the resulting score distributions are less certain.

Figure 13 shows the confidence scores of the inferences obtained with the DNN trained model using the “null” dataset.

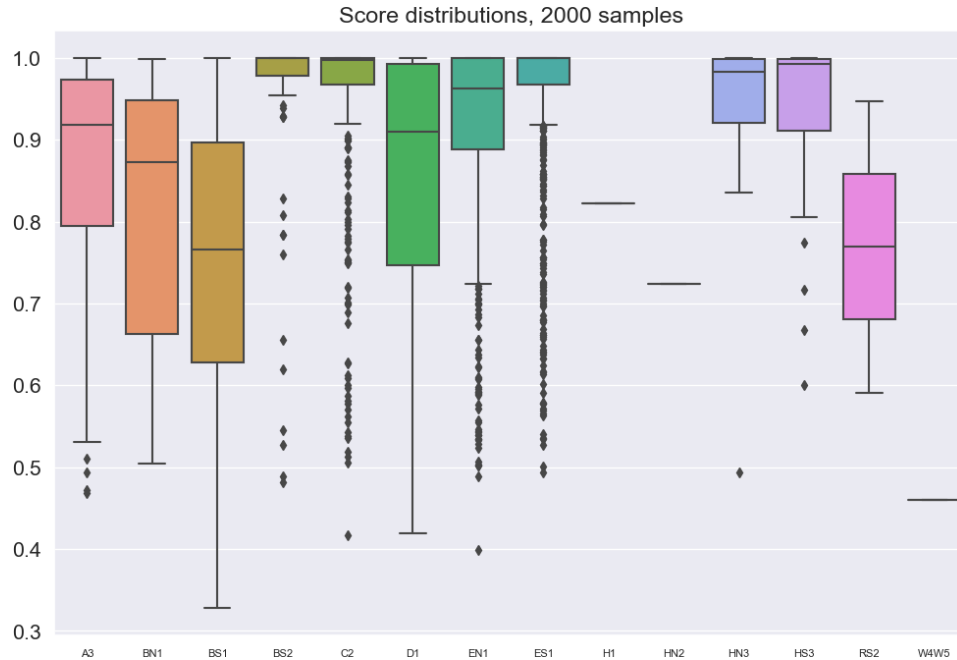


Figure 13. Distribution of confidence scores for the DNN trained on dataset 3 (16 samples observed over an 8-day observation period, uncertainty added) and evaluated using the “null” dataset (2,000 examples).

In this case, there are no true orbit labels, since the features were designed to be ‘out of orbit family’. **Figure 13** shows the confidence score distributions for each of the predicted orbit families for all 2,000 observations. In this case, there were only 14 families predicted. These results show that the DNN trained model has a high confidence (mean > 0.9) for 8 of these orbit family predictions. Ideally, we desire the confidence scores to be low for all samples in the “null” dataset. This is not the case for the ‘null’ samples resulting in high confidence scores. In other words, the DNN model is either: poorly discriminating observations that are not members of some of the cislunar orbit families or; some of the “null” examples are not really that different from true cislunar periodic orbits.

With regards the autoencoder, we trained the DNN-AE with dataset 1 and obtained an overall validation accuracy of 95%. This is slightly less than the performance obtained without the denoiser (97%) but is expected due to the additional loss term for denoising (2). The denoiser performed very well when the feature set was “standardized” (zero mean and unit standard deviation) and achieved an SNR improvement of 42X (32.5 dB). However, we have not yet been able to achieve a similar result when the feature set was not “standardized”.

The RFC was also trained and evaluated on datasets 1, 2 and 3. The number of trees and max depth in the RFC were set to 25 and 30 respectively as a result of a hyper-parameter optimization. The number of observations was varied by subsampling: [16, 8, 4, 2, 1] for the 4 features of interest: RA, DEC and their rates. For subsamples of size 16, all 16 observations within a simple were selected from the dataset. Given a subsample of 8, every other observation was selected. For subsamples of 4, the 1st, 5th, 8th, and 12th observation was chosen. Given a subsample of size 2, the 1st and 9th observation were selected, while a subsample of size 1 selected the 1st observation. Results are shown in the following Tables.

Table 7. Validation accuracy versus sample size for an RFC trained on dataset 1 (16 observations at the same time). Observation sites: Azores and Maldives. Data integrity: noise added. Feature set: RA, DEC, RArate, DECrate.

site\samples	16	8	4	2	1
Azores	0.956	0.953	0.951	0.939	0.902
Maldives	0.913	0.912	0.908	0.891	0.819

Table 8. Validation accuracy versus sample size for an RFC trained on dataset 2 (16 observations over a 2-hour period). Observation sites: Azores and Maldives. Data integrity: noise added. Feature set: RA, DEC, RArate, DECrate.

site\samples	16	8	4	2	1
Azores	0.960	0.959	0.959	0.955	0.952
Maldives	0.908	0.915	0.921	0.920	0.914

Table 9. Validation accuracy versus sample size for an RFC trained on dataset 2 (16 observations over an 8-day period). Observation sites: Azores and Maldives. Data integrity: noise added. Feature set: RA, DEC, RArate, DECrate (4 features).

site\samples	16	8	4	2	1
Azores	0.992	0.992	0.992	0.990	0.961
Maldives	0.993	0.992	0.992	0.989	0.965

Table 10. Accuracy versus sample size for an RFC trained on dataset 2 (16 observations over an 8-day period). Observation sites: Azores and Maldives. Data integrity: noise added. Feature set: RA, DEC (2 features).

site\samples	16	8	4	2	1
Azores	0.986	0.985	0.981	0.962	0.557
Maldives	0.985	0.984	0.980	0.951	0.559

Overall, the RFC performs very well across all three datasets. The loss in performance is obvious as the sample size is reduced. The 2-hour observation data performs worse than the 8-day data and is in agreement with the DNN results. The RFC did not perform as well as the DNN for dataset 1 (16 samples all at the same time) when noise was added (96% vs 97%) but the difference is minute.

An RFC model trained on dataset 3 was also evaluated using the “evaluation” and “null” datasets. These results are shown in **Figure 14** and **Figure 15** for the “evaluation” and “null” datasets respectively.

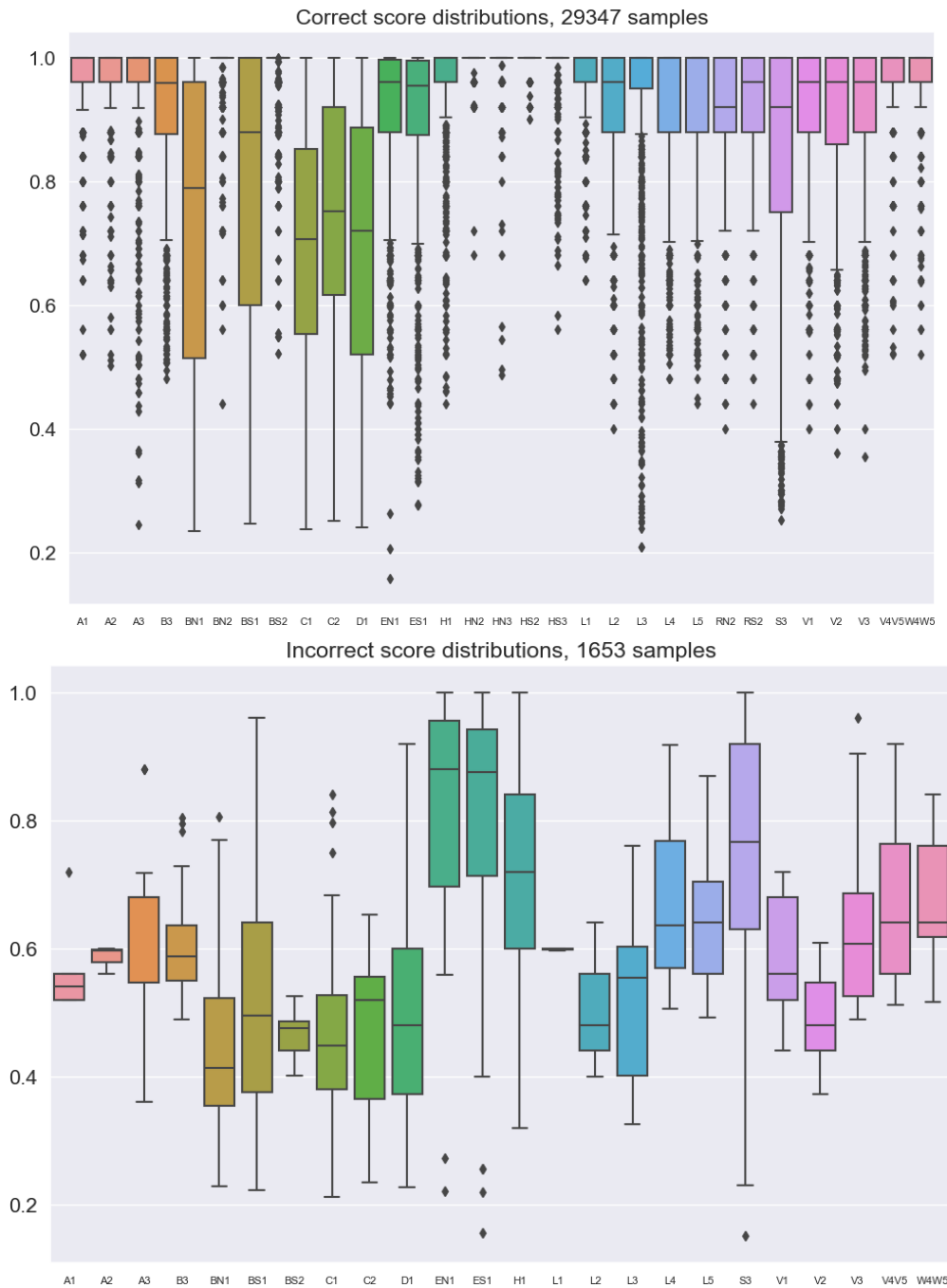


Figure 14. Distribution of confidence scores for the RFC trained on dataset 3 (16 samples observed over an 8-day observation period, uncertainty added) and evaluated using the “evaluation” dataset (31,000 examples, 1000 examples/orbit family). Top graph: Correct orbit family prediction; Bottom Graph: Incorrect orbit family prediction.

These results show very good performance with respect to high confidence score for correctly predicted orbit families with low spread of distribution. Of the 31,000 examples in the ‘evaluation’ dataset, 29,347 were correctly classified for an overall accuracy of 0.95. There were 5 of the orbit families whose mean confidence score was 0.9 or less: BN1, BS1, C1, C2 and D1. These same families have the lowest mean confidence score for the incorrect orbit family classification (Bottom Graph) .

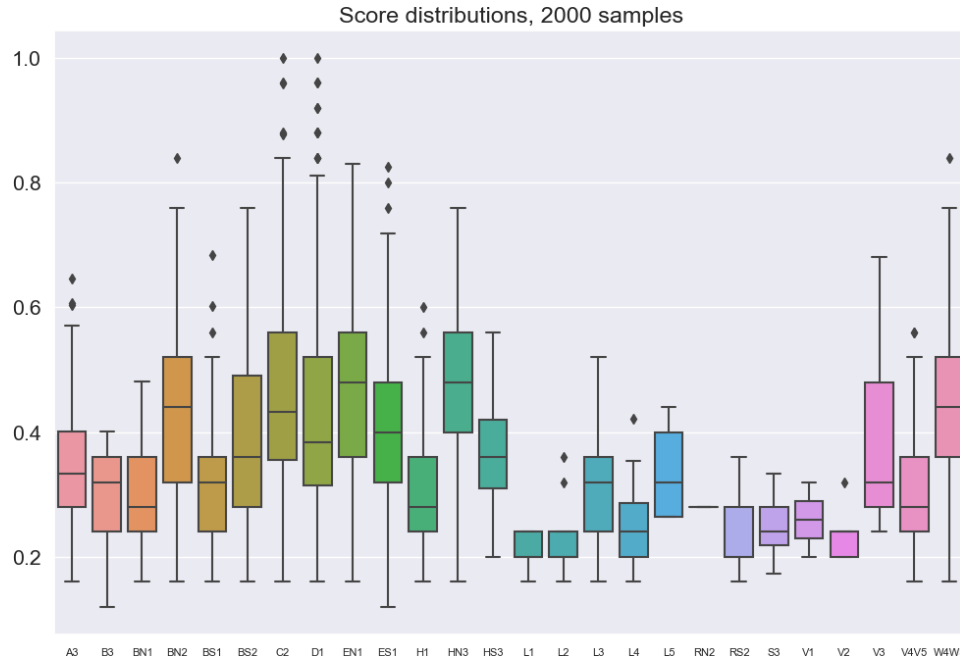


Figure 15. Distribution of confidence scores for the RFC trained on dataset 3 (16 samples observed over an 8-day observation period, uncertainty added) and evaluated using the “null” dataset (2,000 examples).

The RFC performed well when evaluated with the ‘null’ dataset. Average scores are generally quite low with an average confidence of about 0.35 and the distribution of scores has low variability. Only 26 of the 31 cislunar orbit families were classified. Missing orbit families were A1, A2, C1, HS2, HS3.

In this section, the DNN and RFC were evaluated with regards to how well they can correctly classify observations of satellite trajectories into cislunar orbit families. Classifying an observed trajectory into a cislunar periodic orbit family can be quite beneficial in IOD.

Both Classifiers performed quite well using only 4 features: RA, DEC, RArate, DECrate, even with a fairly large amount of added uncertainty. Of the two classifiers, the RFC appears to provide the highest accuracy over all conditions tested.

7. Visualization

A cislunar orbit visualization tool was written in the Unity Framework. The user can select a cislunar periodic orbit family and it is drawn on the screen along with the Earth and Moon. A sample feature set (16 samples over an 8-day observation period) from the validation set on which the DNN was trained is then selected and displayed. The predicted (inferred) cislunar family is shown in the upper left dialog box along with a confidence score and the true cislunar family. See Figure 16 below for an example of the display.

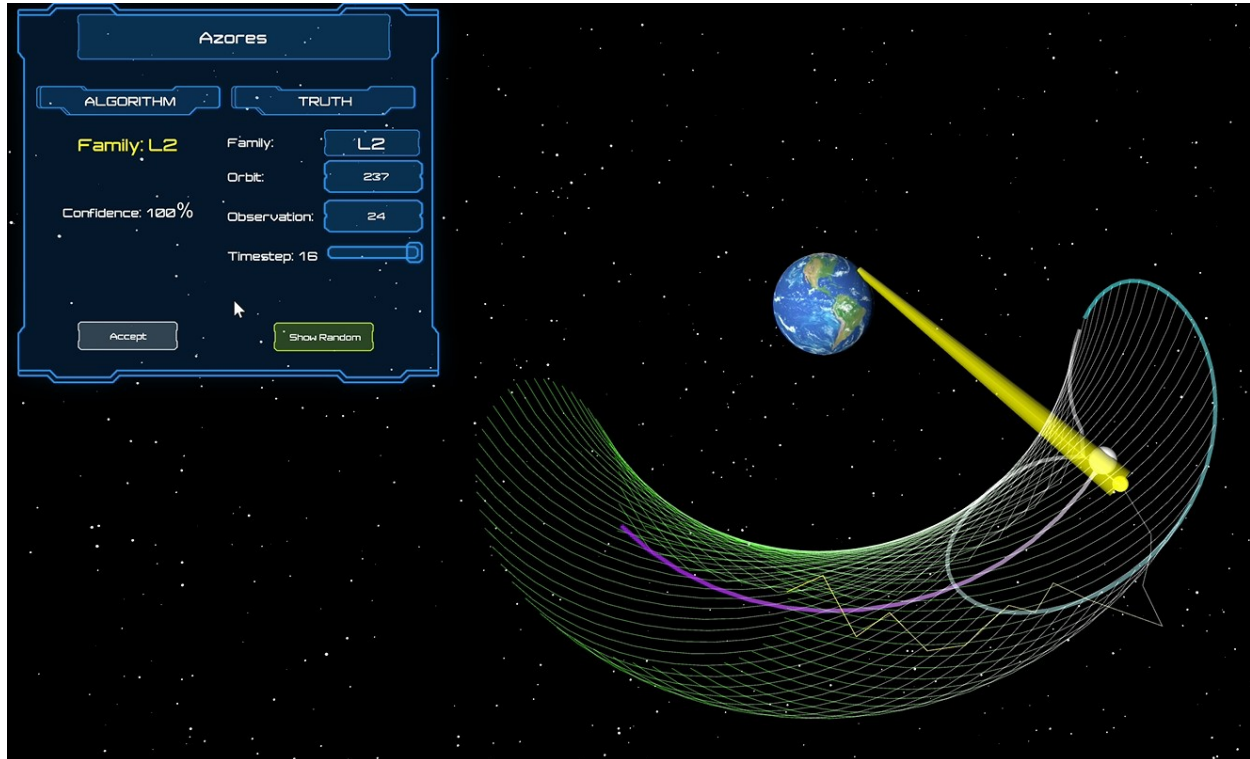


Figure 16. Visualization of a member of the L2 orbit family. Green lines: pristine L2 orbit over various times in inertial space; Gold spheres: sequence of 16 noisy observations over 8 days; Upper left dialog: true and DNN predicted orbit family; confidence score; Azores observation site.

8. CONCLUSIONS

Machine-learning techniques are successfully applied to classifying observations of varying size as belonging to one or none of the elemental periodic orbit families. This determination is often not possible with a single or a small number of observations (e.g. less than six) using conventional estimation techniques. The implication is that IOD algorithms can perform better if informed by these machine-learning algorithms to help constrain the admissible regions for track fitting.

Future work includes a more comprehensive investigation into other dynamical systems structures (e.g. quasi-periodic orbits, invariant manifolds, homo- or hetero-clinic connections) to broaden the classifiers ability to place membership into other groupings. A sensitivity study

should also be conducted to test the robustness of the classifiers on “nearby” solutions, corner cases, or on different observing conditions and types.

Future work will focus on developing models that: perform well under varying lunar and solar phases; explore the use of ensembles of all the models (via a majority vote) to improve overall accuracy of classifying an observation as a member of a cislunar family.

9. REFERENCES

Chow, C. C. et al., 2020. Cislunar periodic orbit families and expected observational features. *The Advanced Maui Optical and Space Surveillance Technologies Conference*, 15-18 September.

He, K., Zhang, X., Ren, S. & Sun, J., 2015. Deep residual learning for image recognition. *arXiv:1512.03385 [cs]*, December.

Doedel, E. et al., 2007. Elemental periodic orbits associated with the libration points in the circular restricted 3-body problem. *International Journal of Bifurcation and Chaos*, 17(8):2625-2677.