Toward using Machine Learning (ML) models for data association and maneuver classification of Resident Space Objects (RSO's)

Triet Tran
Cornerstone Consulting & Services, LLC

Anthony N. Dills
L3Harris Corp.

ABSTRACT

The ability to perform near real-time data association and automatic detection and classification of resident space object (RSO) maneuvers is highly desirable. The potential benefits include anomaly resolution and change detection that may improve the accuracy of the orbital state of Earth orbiting satellites and debris.

A challenging aspect of Space Situational Awareness (SSA) and Space Traffic Management (STM) is detecting an RSO maneuver based solely on observations and current knowledge of the orbital states. Modern techniques have been developed by STM researchers using statistical filtering techniques with various degrees of success. Artificial Intelligence/Machine Learning (AI/ML) techniques have seen significant growth in recent years and pose viable approaches within the space domain's solution space to augment, supplement, and potentially replace traditional methods with varying degrees of performance.

The contribution of the present work is to demonstrate the feasibility of using AI/ML to perform data association and maneuver classification of the RSO's in the geosynchronous (GEO) orbit regime. Pattern of Life (POL) of the maneuvering RSO is the key to the success of the AI/ML model to recognize whether a set of optical measurements can be associated with an RSO candidate and whether a maneuver type can be classified for that RSO based on the same optical measurement set. This work presents an autoencoder (AE) to perform data association using historical maneuver data for a particular RSO as observed by a specific ground-based optical sensor. The same data is also applicable to identifying and training an optimized classification model by applying an open-source genetic programming technique known as Tree-Based Pipeline Optimization Tool (TPOT). The presented analysis used precision ephemerides publicly available from the National Oceanic and Atmospheric Administration (NOAA) and Federal Aviation Administration (FAA) for GEO earth orbiting satellites. The data used for assessing AI/ML candidate models are the simulated ground-based optical measurements on RSO's whose POL of maneuvers is derivable from publicly available ephemerides.

Results of this work point to an alternative approach of data association and maneuver classification. The results also demonstrate an approach for real-time association that could be implemented at an observing ground site to help reduce orbital state uncertainties in collaboration with existing astrodynamics filtering techniques. The results also identify the most prominent features of the data that reduces the dimensionality within classification model. Finally, an approach to automatically label maneuvers in each data set of observations was developed and could provide a utility to the greater community.

The data association and the classification results each achieved better than 90% prediction accuracy based on validation during the training phase of the ML models. Future work will include model tuning, examining data diversity, assessing model prediction of real observation data, exploring the use of model ensemble to increase robustness, and addressing applicability to the more challenging aspect of data association for closely spaced RSOs.

1. INTRODUCTION

A desirable operational capability is the ability to perform near real-time data association and maneuver detection of an RSO using observations from ground- and space-based sensors. The potential benefits include anomaly resolution and change detection that will improve the accuracy of the orbital state of the space objects. Maneuver detection using optical observations is very challenging for several reasons: often the observations are sparse, data association causes mis-tagging of the RSO, and orbital uncertainties often mask the effect of maneuvers when the magnitude are small, such in as low-thrust maneuvers [1].

In the past decade there have been significant contributions in the area of data association and maneuver detection using modern filtering techniques and optimal control algorithms (for example, see [2] [3] [4] [5] [6] [7] [8]) that can be coupled with ML models to form a complete self-contained tool for deployment at the sensor site for near real-time maneuver detection. More recently, contributions in maneuver detection using ML modeling techniques (such as in [9] [10] [11]) show promising results. In [9] for example, Abay et. al. illustrate that Generative Adversarial Networks can learn the normal behavior of the resident space object under simulated natural forces and detect maneuvers as anomalies. Whereas, similar to the current effort, [11] uses an unsupervised Interval Similarity Model to establish correlated patterns of life between astrometric observations and associated maneuvers.

In this paper we demonstrate the feasibility of an ML model to perform data association and to classify maneuvers of a well-behaved GEO RSO into North-South (NS), East-West (EW), and Radial (RD) station-keeping maneuver. The ability of the automated ML to choose the best classifiers is demonstrated using TPOT. The data association capability is demonstrated using the well-known autoencoder that has been used for different purposes in the Data Science community ([12] [13] [14]). In this work, it is theorized that the pattern-of-life observed by a tracker on a given RSO is unique, the simulated data generated for training of the model is for a specific ground-tracker collecting observations on a specific RSO. The assumed POL uniqueness is explored with a different ground-tracker on the same RSO (or with the same ground-tracker and a different RSO) to validate the uniqueness in the AE characteristics and thus the tracker-RSO pairing.

The layout of the rest of the paper is as follows: Section 2 reviews the background on autoencoder and TPOT, Section 3 discusses the features of the training data for the ML models, Section 4 presents the result of training and prediction of the models, and Section 6 provides conclusion and future work.

2. BACKGROUND ON AUTOENCODER AND TPOT

Autoencoder

AE is a type of unsupervised Neural Network (NN) consisting of two primary elements: encoder and decoder. Figure 1 illustrates the nominal architecture of the AE. During the encoder stage, the AE will try to extract the most relevant features of the input data and stored them at the coded layer. Then the decoder uses the code layer as input to reconstruct the output layer with the same dimension as the input layer. The loss function for any AE is the error between the output and input layer. The most common loss function is the Mean Square Error (MSE) or the Mean Absolute Error (MAE). It is this reconstruction error that is used by the optimizer to search for the set of weights and biases for the AE neural network. The optimizer is usually the stochastic gradient descent; therefore, each train/validation cycle will produce different results. The resulting statistics must be obtained via multiple trials run to collect mean and standard deviation of the performance metrics. This will be done in future work to document the statistical characteristics of the stochastic search.

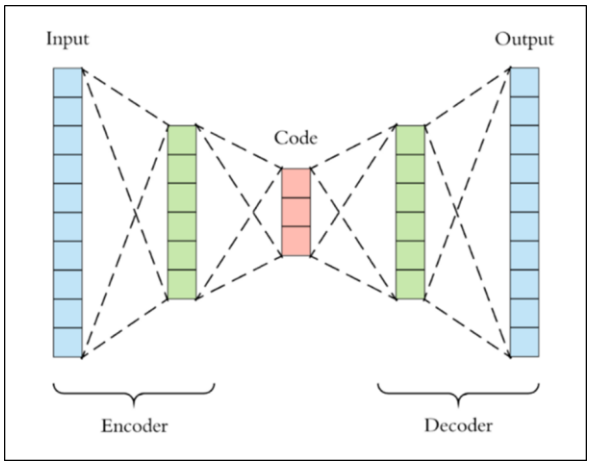


Figure 1. Autoencoder Architecture

The code layer has fewer elements than the input layer, and therefore it is used as a data compressor for efficient transmission of data that has acceptable loss in the compression process. In this study, the AE is used as a binary classifier for data association where the optical observation data belonging to the Tracker-to-RSO line-of-sight geometry will have the reconstruction error to be below some specified threshold ("True"), while data not belonging the pair will exceed the same threshold ("False").

An example of the reconstruction of a Modified National Institute of Standard and Technology (MNIST) database image [15] of a hand-written number using AE is shown in Figure 2.

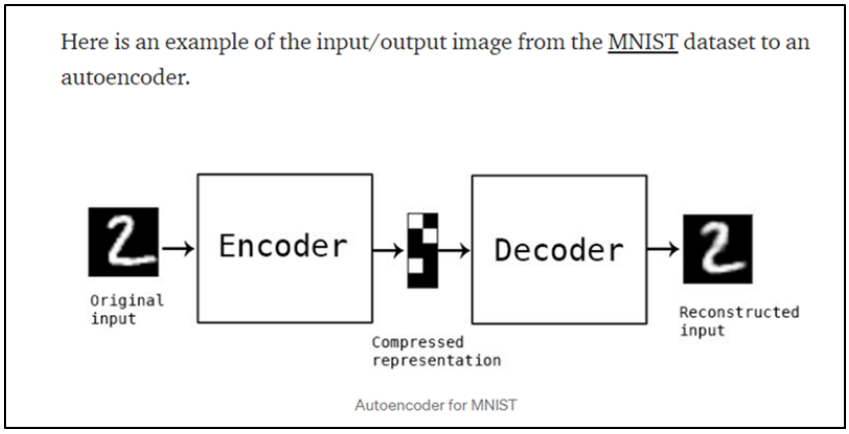


Figure 2. Reconstruction Process of an Autoencoder on a Hand-Written number [15]

Tree-based Pipeline Optimization Tool (TPOT)

TPOT uses genetic programming executing a series of "pipeline" operations to determine an optimal classifier in the scikit-learn library of ML models [16]. An example TPOT pipeline is shown in Figure 3. The TPOT genetic algorithm optimizes a loss function to determine the best-performing model. The automated feature engineering (e.g. Figure 4) is at the heart of exhaustively exploring the dimensionality of the input feature vectors to yield high performance.

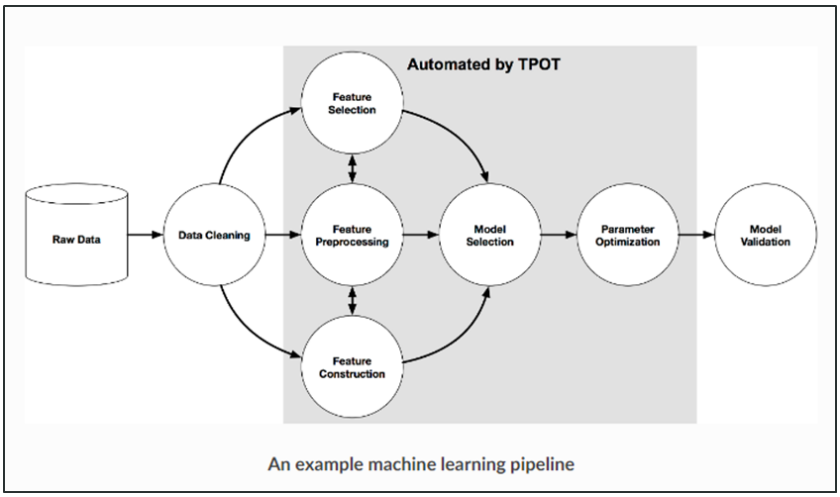


Figure 3. Example of Machine Learning Pipeline [16]

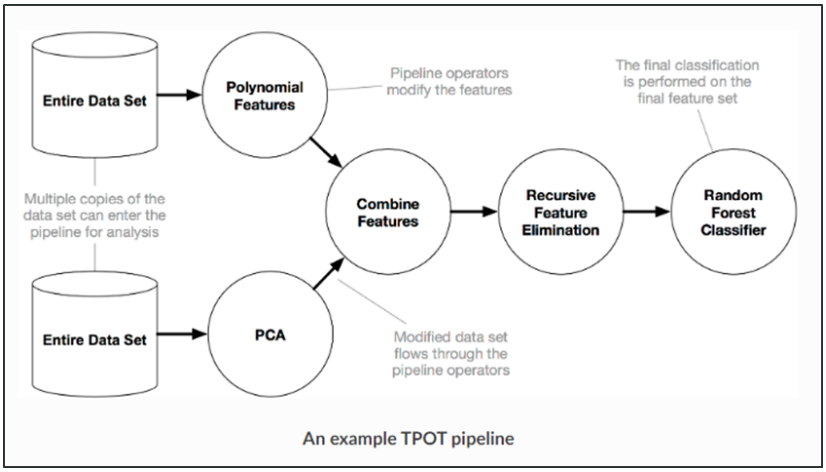


Figure 4. Example of TPOT Feature Dimensionality Reduction [16]

Configuration for TPOT

As with any ML model, hyper parameters must be tuned for optimal performance. TPOT's genetic algorithm approach will tune each candidate ML model; however, the level of TPOT searching of those models and parameters is governed by a set of configuration parameters. The parameters used in this effort at listed in Table 1.

Table 1. TPOT Configuration Parameters

Parameter	**Value**	**Parameter**	**Value**
'generations' (iterations)	20	'verbosity' (runtime output level)	2
'population_size' (individuals)	30	'scoring'	F1_weighted
'cv' (number of cross validations)	10	'n_jobs' (parallelization)	-1 (for all CPUs)
'random_state' (seed)	42	'config_dict' (built-in configurations)	None

Once TPOT has been trained, top performers can be examined as shown in Figure 5 which illustrates an example of the top-performing classifiers and their corresponding quartile plot for the cross-validation score. This graphic helps the analyst to visually compare the relative performance of the models and may guide the final model selection based upon not only the score but also the spread of that score.

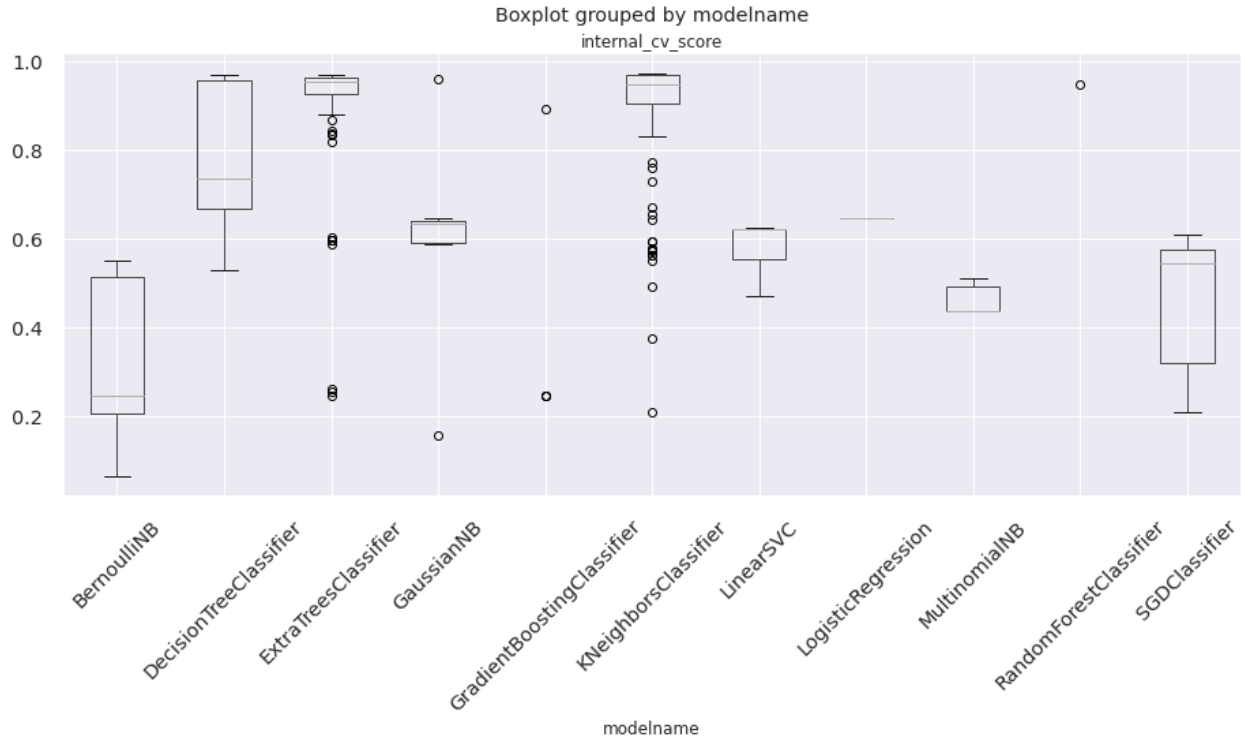


Figure 5. Quartile plot of TPOT classifiers

3. GENERATING FEATURE DATA FOR MODEL TRAINING

Selecting the relevant data features (e.g. data wrangling) for training the model is one of the most time-consuming aspect of preparing to applying AI/ML models to specific domains. Once the features were defined, the data was split into training and test sets. The tests set was left out of all training and cross-validation to preserve its independence to evaluate the performance of the models and better reflect operational scenarios.

The initial feature set includes a variety of parameters associated with the observations (measurements) and the orbital elements derived from the knowledge of most up-to-date orbital state as stored in the RSO catalog. However, as the issue of associating the optical data from the ground-based optical tracker drove the finding that one cannot assume any orbital knowledge associated these "raw" measurements. Therefore, the catalogued orbital elements can only be used as a reference against which the raw (i.e. un-associated) observations are to be constructed into training data for the ML models. Also hypothesizing that the geometric relationship between the tracker and RSO is unique, an ML model to predict association must also be unique for that ground-tracker and specified RSO. Using the illustration in Figure 6, consider the scenario where a specified ground-tracker collects optical data over a given duration that may have multiple snapshots of the RSO's against a star background. These optical observations are then reduced into pairs of right-ascension/declination (RA/DEC) data points for the observed epoch. The ML model, when properly trained, should associate the RA/DEC data pairs to a set of RSO's that may have crossed the field-of-view (FOV) during that observation time window.

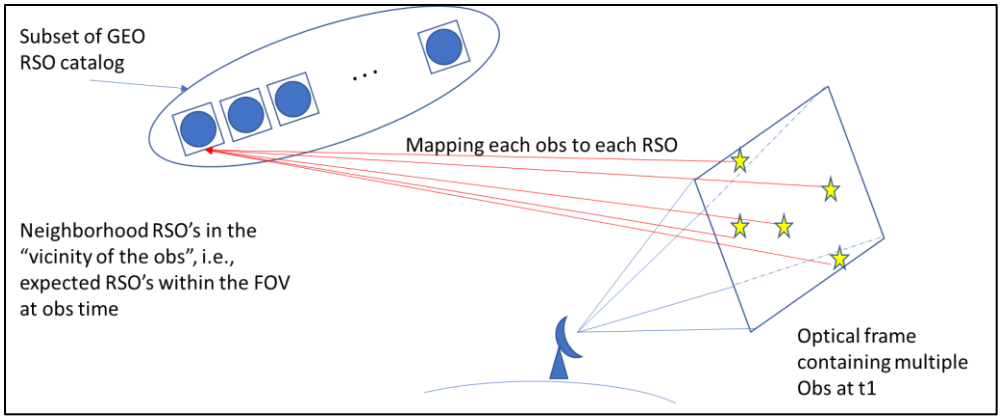


Figure 6. Data Association Concept of Operations

As noted in the results section, as the ML AE program processes the optical observations to predict whether the observed RA/DEC observation belongs to an RSO that may have crossed over the FOV of the tracker's site, a strength-of-association metric can be generated for each Tracker-to-RSO association that can be used as for filtering the observations. This metric can be used as a metric to assess the performance of the predictions by the AE.

Due to the uniqueness discussed, the resulting features used to train the models include the difference in both the spherical angles and the component of the line-of-sight (LOS) unit vectors in various coordinate frames including topocentric, earth-fixed, earth-centered inertial, FOV and measurement residuals. For a detailed description of the parameters derived from optical data, please see the Appendix. In all, there were 28 resulting features for each observation record for the Tracker-to-RSO pair for optical data.

When computing the feature set, the precision ephemerides of Galaxy15 for the entire year 2016 was used which include 52 actual maneuver events of various maneuver types as shown in Table 2. From the perspective of training data, 58 maneuvers of various types are not statistically significant for ML model to train on. Further note that only 3 radial maneuvers were executed for that year. The effect of unbalanced data shows up in the performance of the model as seen in the Section **Error! Reference source not found.**. For this reason, we generated multiple records of simulated orbit states where orbital errors in both position and velocity components sampled from a normal distribution were injected into the orbit states to ensure statistical significance for the training data.

Table 2. Maneuver Events for Galaxy15 in 2016

Maneuver Event Type	Number of Occurrences
East-West (EW)	27
North-South (NS)	28
Radial (RD)	3

Automatic Data Labeling of Maneuver Events

With the precision ephemerides, a simple technique was developed to label the maneuver events automatically. Figure 7 illustrates the maneuver detection technique using consecutive records within the ephemeris data. The open source OrbDetPy [17] was also used as an orbit propagator to aid in the maneuver detection. This detection technique uses the state at one record, say at time t_1, to seed the orbit propagator as an initial state which is propagated to the time of the next record, t_2. The velocity vector from the propagated state is subtracted from the velocity vector of ephemeris data record at time t_2 to obtain the delta-V vector which is then transformed into a radial, in-track, and cross-track (RIC) coordinate frame (also known as UVW coordinates in some domains). Each component of the delta-V in the RIC frame is examined to determine whether a maneuver is in-track, cross-track, or radial. Based on the observed noise in the ephemeris, a threshold of 2.5 cm/s was used as a lower limit, above

which a maneuver event was labeled. These components also correspond to a station-keeping maneuver as EW, NS, and RD maneuver, respectively, for an RSO in Geosynchronous orbit.

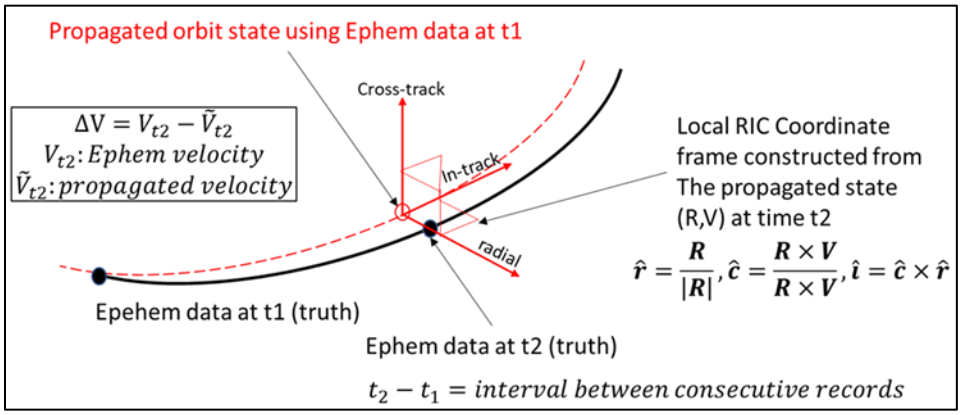


Figure 7. Maneuver Detection using precision ephemerides

When computing ground-based optical measurement and residuals, we used the pre-maneuver state 24 hours prior to the maneuver event to compute the expected optical measurement. The truth orbit (via precision ephemerides) was used to compute actual optical measurements including the effect of normally distributed measurement noise.

Finally, each feature record was labeled with (0,1,2,3) corresponding to no maneuver, EW, NS, and RD maneuvers, respectively. For example, all measurement records occurring prior to the maneuver event are labeled with '0', and any optical measurement occurring post maneuver is labeled as (1,2,3) depending on the maneuver event type. The computed optical measurement for a fictitious ground-station were generated as long as that station had visibility to the RSO of interest during the maneuver events.

Injection of Orbital Errors into the Initial State

Orbital state vectors inherently contain uncertainty that is characterized by a covariance matrix that can be generated through the traditional batch or sequential estimation processing of the observations. Normally distributed position and velocity errors to the Cartesian State were added to estimated orbital errors. Large orbital errors were observed to adversely affect the ML model maneuver detection performance. Initially, this work set the position error to zero mean and standard deviation of 2 km, and the velocity error to zero mean and standard deviation of 0.2 m/s. The nominal state was taken at 24 hours prior to the maneuver event. The computed ground observations were collected every 5 minutes starting with 3 hours before the maneuver event and ending at 12 hours after the maneuver event. For the Galaxy15 RSO, there were typically 30 to 40 maneuver events per year. Three data collections of optical measurements per event were generated each time by randomly injecting orbital errors into initial state 24 hours before the maneuver event time.

4. RESULTS

Data Association with Autoencoder Model

The data association AE was built with the encoder and decoder each having two hidden layers as shown in Figure 8. The input layer had 28 elements, followed by two hidden layers for the encoder with 20 and 16 elements respectively, and 10 elements for the code layer; the decoder was kept symmetric with respect to the encoder. The AE was trained for 400 epochs with the history of the training shown in Figure 8. The behavior is this training indicates no over-fitting as both training and validation tracking each other closely.

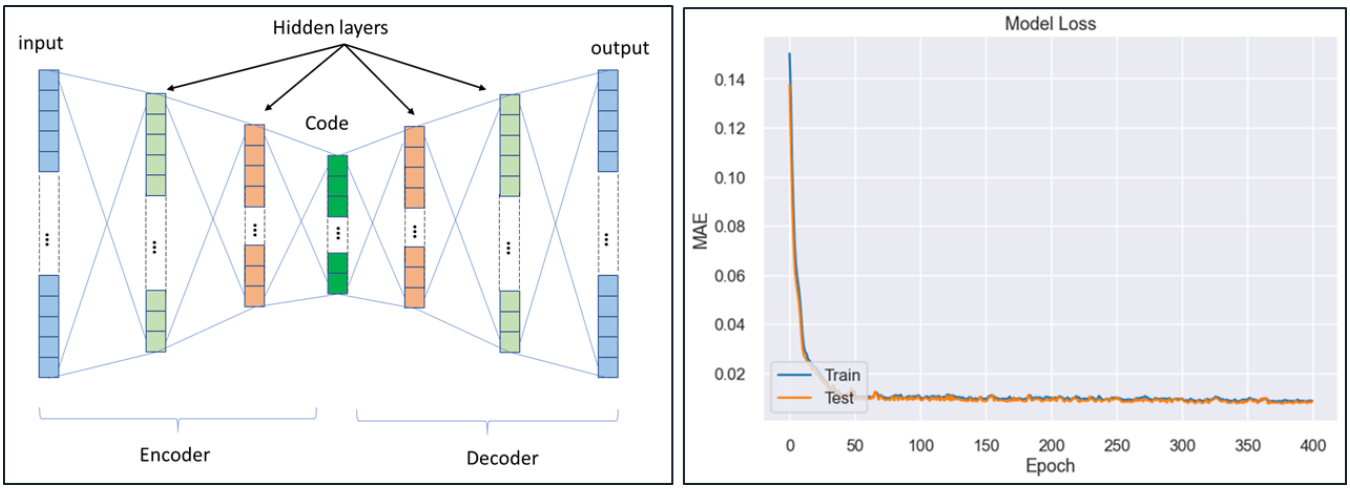


Figure 8. Architecture and Training History Curve for the Data Association Autoencoder

Once trained, the AE can be used to predict the reconstruction error assuming that any input that has the same number of the elements as the training data (i.e., dimensional matching). To be used as a binary classifier, a population inclusion value was selected. Initially, the inclusion value that contains 98% of the samples was chosen and resulted in a MAE threshold value of 0.0182.

Figure 9 shows the histograms of the reconstruction MAE of the training data for the Galaxy15 2016 maneuver history. The data association AE model as trained was used to predict the MAE when applied against the Galaxy15 2017 and 2018 data (orange hue in Figure 9). It was evident in Figure 9 that most of the samples for the Galaxy15 maneuver data in 2017 and 2018 lie below the 98% inclusion value. The resulting prediction accuracy was 90.6% for 2017 data and 91.6% for 2018 data. Operationally, any optical observation for given ground optical tracker can have its MAE predicted via the trained model. Thus, any records that fall below specified inclusion value is assessed as belonging to this Tracker-RSO pair; otherwise, they do not. This is the essence of the data association operations using ML model such as the autoencoder.

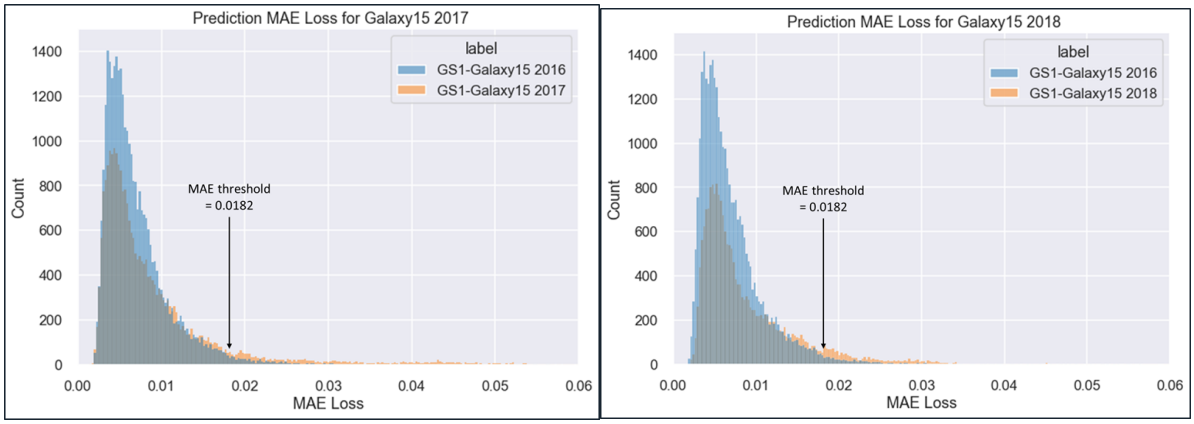


Figure 9. Histogram of Reconstruction MAE using Autoencoder for Galaxy15: Trained with 2016 data (blue) and Predicted with 2017 data (orange)

As previously mentioned, the input feature parameters set is unique for each pair of Tracker-RSO with respect to the data association AE. Furthermore, as shown in Figure 10 the MAE from the same ground optical tracker will have a different MAE loss for a different RSO than the one originally used to train the AE.. The unique characteristics of the MAE for a different Tracker-RSO pairs demonstrate the utility of the AE to perform binary data association for any optical observations.

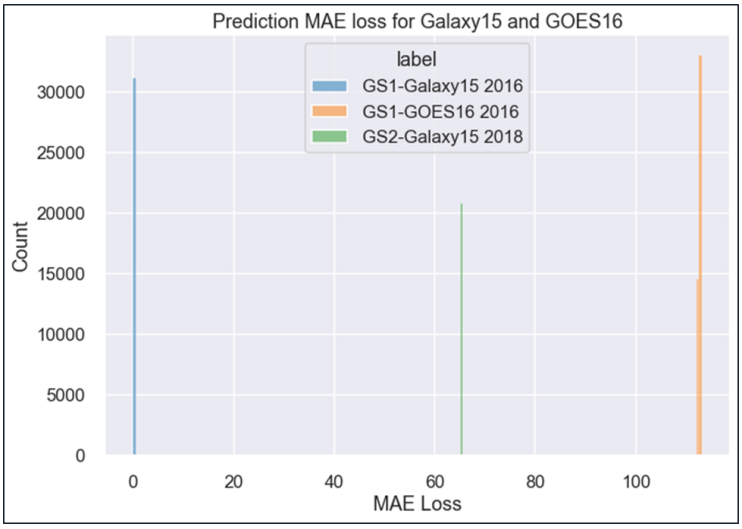


Figure 10. Predicted MAE Reconstruction Loss histograms: light-blue is for GS1- Galaxy15 pair; light green is for the same RSO (Galaxy15) but tracked by a different ground optical tracker (GS2); and light orange is for the same ground optical tracker but observing a different RSO (GOES16).

Binary Maneuver Classification using Autoencoder

As mentioned above, the Galaxy15 data includes observations for both non-maneuver and post maneuver events. It was desired to explore the performance of the AE to classify observations into non-maneuver and maneuver by training on 75% of the non-maneuver observation 2016 data. Subsequently, the trained model was then used to predict the MAE on the remaining 25% maneuver data within the same year. Figure 11 shows the histogram of the MAE using non-maneuver data. Selecting the threshold based on inclusion of 98% of the data samples resulted in a prediction accuracy of 97.75%.

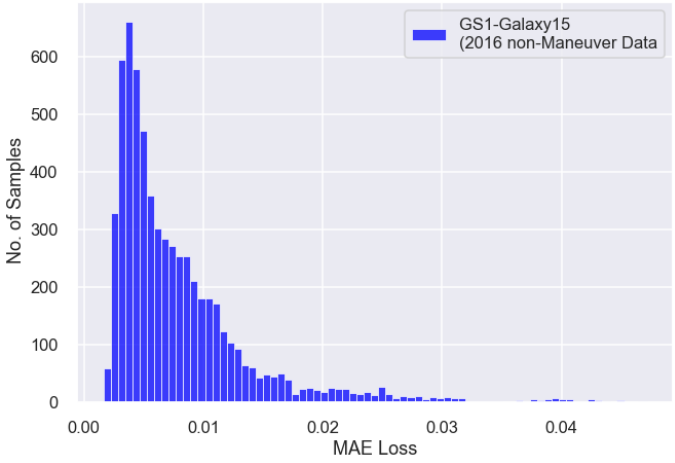


Figure 11. Prediction of MAE reconstruction error trained on Non-Maneuver Data.

The histogram of the prediction MAE for maneuver using the AE binary classifier trained with non-maneuver data is shown in Figure 12. For this prediction, the accuracy is only about 58%. The threshold selected from the 98% sample inclusion is found to be 0.038. Based on these findings, the binary maneuver classifier using autoencoder does not perform as well as the AE for Data Association. Note that tuning has not been performed for the AE models.

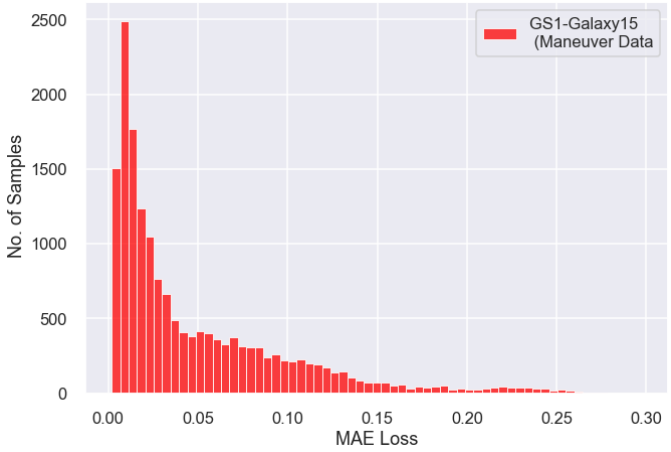


Figure 12. The prediction of the MAE for maneuver data using AE ML trained with non-maneuver data

Binary Maneuver Classifier using TPOT/ExtraTree classifier

As an alternative to the Autoencoder presented above, TPOT was used to determine the best binary classifier between maneuver and non-maneuver. In this case, a maneuver type 1, 2, or 3 was labeled True while a maneuver type 0 was labeled False. Using the same parameter search window as before, TPOT identified the scikit-learn's ExtraTreeClassifier as the best performer [18]. The ExtraTreeClassifier was then tuned over 25,000 iterations, outside of TPOT, with RandomizedSearchCV function to determine the optimal hyperparameters (see Table 3) to achieve a 0.91 F1 macro score and a 0.91 receiver operating characteristic area under the curve (ROC AUC) (statistical results shown in Figure 13) and a resulting feature saliency shown in the left side of Figure 14.

Examining the top features, each of the first 9 indicate noticeable contributions towards importance of the model performance. After 9, the features tend to level out in performance. An interesting finding is when retrained with only those 9 features, their ranking of importance, as seen in the right side of Figure 14, shuffled slightly with a marginal drop in F1 (0.90). The tuned hyperparameters in this latter case resulted in only the "max_features" changing to 9 (as expected), leaving the remaining parameters shown in Table 3 unaltered.

Table 3. Hyperparameter tuning parameters for ExtraTreeClassifier

Parameter	**Short Definition from Scikit-Learn**	**Search Range**	**Optimum**
max_depth	The maximum depth of the tree	3 or None	None
max_features	The number of features to consider	1 – 20 (unless the number of features were less)	18
min_samples_leaf	The minimum number of samples required to be at a leaf node	1 – 9	1
criterion	The function to measure the quality of a split ("gini" for the Gini impurity and "entropy" for the information gain)	gini or entropy	entropy

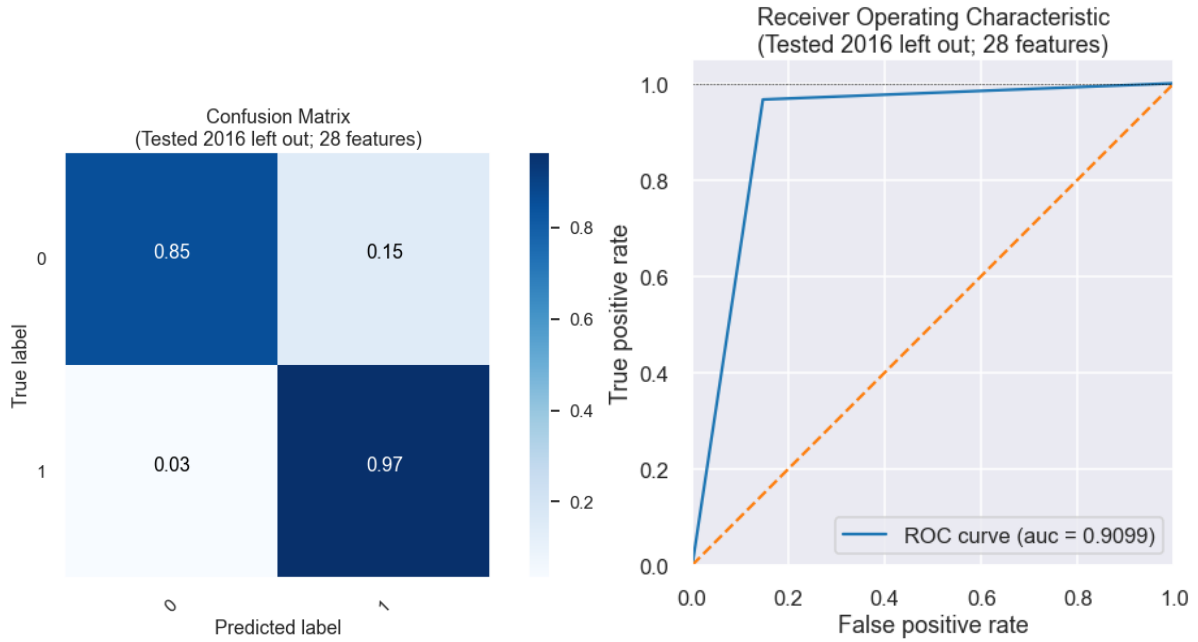


Figure 13. Statistical performance of ExtraTreeClassifier

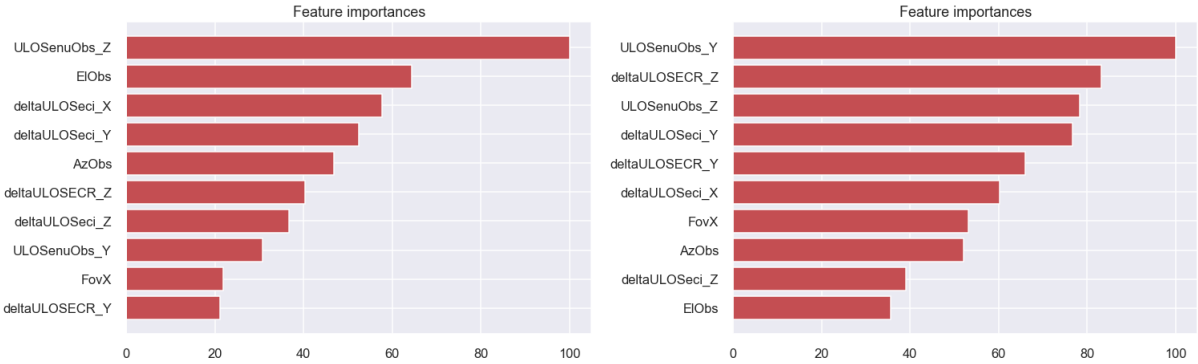


Figure 14. Feature saliency using ExtraTreeClassifier where the left is the ranking using 28 features and the right with the only the top 9 from the left model, yielding an F1 score of 0.91 and 0.90 respectively

The confusion matrix shown in Figure 13 (left) shows the model performance tested on data excluded from the training set for the same year. The 85% to 97% in accuracy performance creates optimism in the model, but degraded in test performance toward 45% and 88% when applied to the same trained model for different years is shown in Figure 15. Still the degradation in prediction performance was not significant in detecting true maneuvers. The accuracy of classifying non-maneuvers degraded into a basic coin toss, i.e., a 50/50 likelihood. Similarly, the ROC AUC decreased to 0.63 and 0.67 for the 2017 and 2018 test years, respectively. It is suspected that the lack of data diversity may have contributed to the degraded performance; an issue to be pursued in future work.

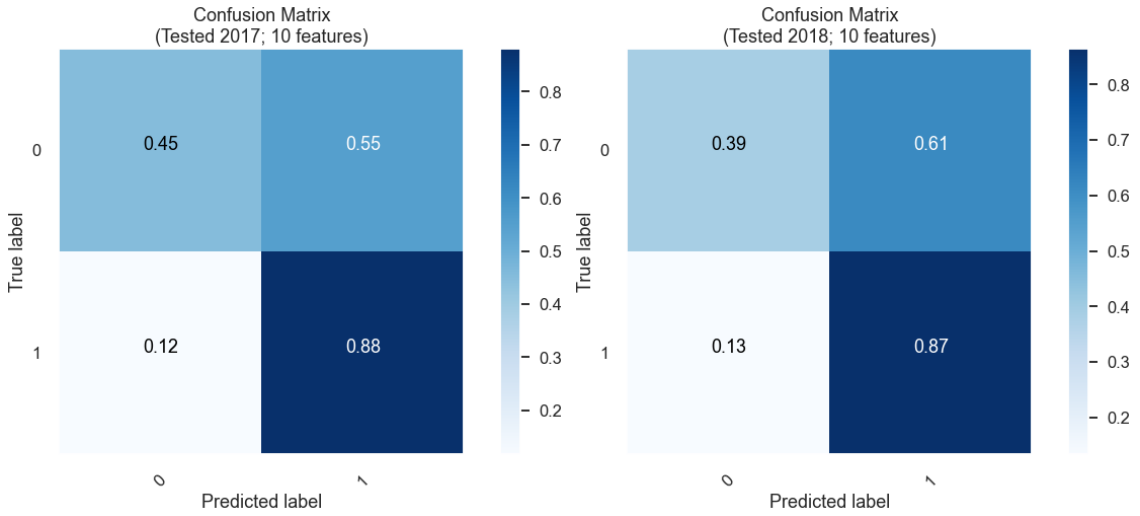


Figure 15. ExtraTreeClassifier binary performance when tested on the next two years following training

Comparison between TPOT and Autoencoder

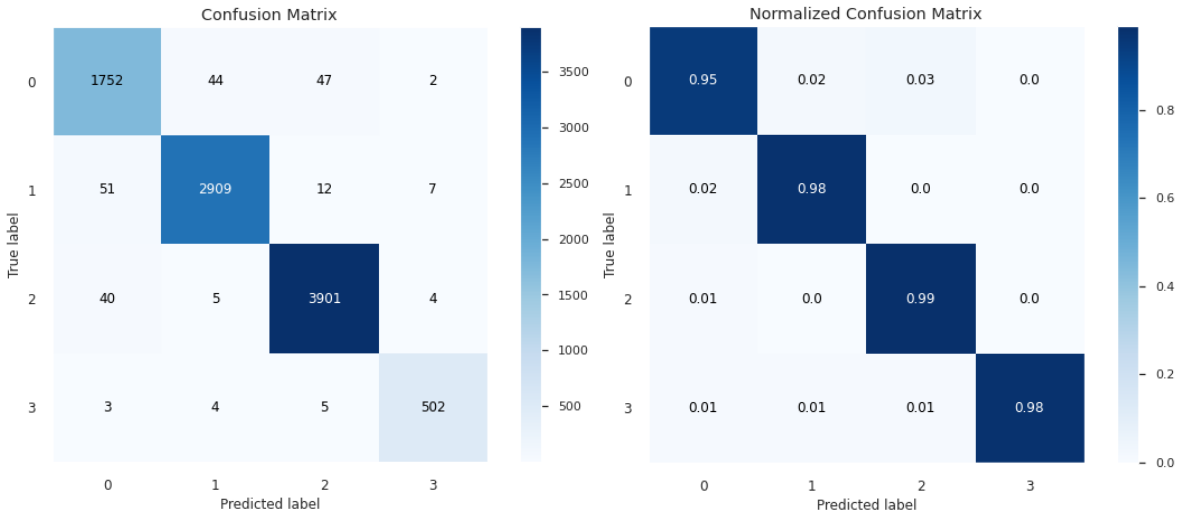
The strengths and weaknesses of each model, TPOT and AE, were compared for binary maneuver classification. As seen in Figure 16, TPOT performed poorly at classifying non-maneuvers yet performed with 87% accuracy at classifying maneuvers. On the other hand, the opposite is true for the AE, meaning that the AE does well (98% accurate) at classifying non-maneuvers but not the other way around. This suggests that to improve prediction performance for binary maneuver classification, a combination of these models is necessary. This is another topic for future work.

		TPOT Binary Classifier			AE Binary Classifier	
Truth	0	0.45	0.55		0.98	0.03
	1	0.13	0.87		0.40	0.60
		0	1		0	1
		Predicted			Predicted	

Figure 16. Comparison of confusion matrix for TPOT and Autoencoder for binary maneuver classification

Multi-class Maneuver Classifier using TPOT

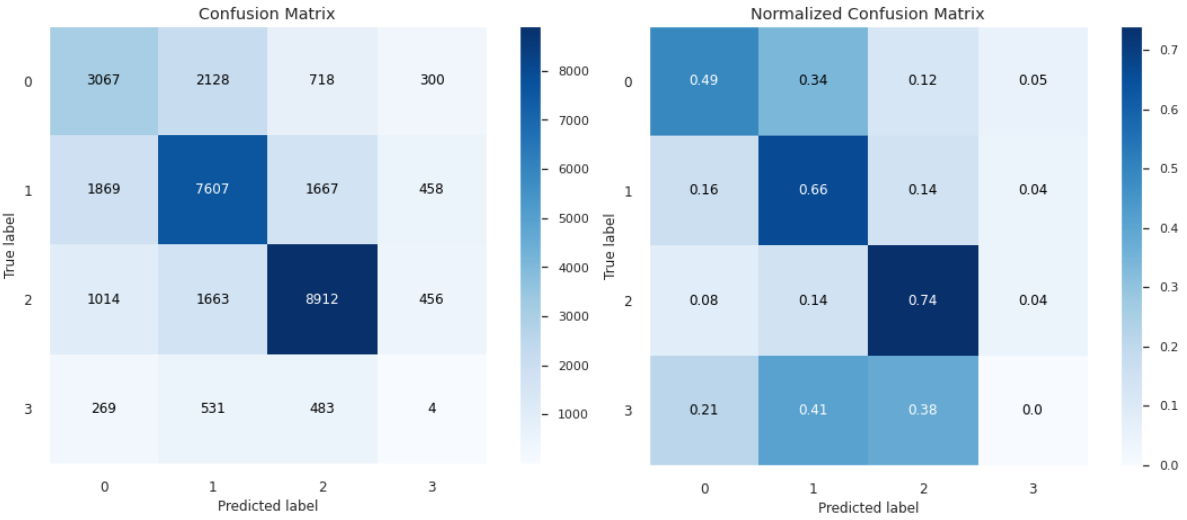
The same training data used for the AE was also used for the TPOT model to predict 4 maneuver classes (None, East-West, North-South, and Radial). In this case, the Galaxy15 2016 data was used to train and validate the model. The data was split into train and cross-validation in the ratio of 25% and 75%, respectively. Once trained, the top model was selected based on the best cross-validation score. The ExtraTreeClassifier was again the top performer among a handful of classifiers models available in scikit-learn library, yielding the performance metrics shown in Figure 17. Overall, the classification accuracy is at or above 95%.



(a) (b)

Figure 17. Confusion Matrix for TPOT Validation Data (GS1-Galaxy15 2017 and 2018 Maneuver Data): (a) Maneuver Category Population Count, (b) Maneuver Category Percentage

The optimized TPOT classifier was then used to predict the maneuver type using the 2017 Galaxy 15 maneuver events yielding the confusion matrix shown in Figure 18.



(a) (b)

Figure 18. Confusion Matrix for TPOT Prediction of 2017 Maneuver Data: (a) Maneuver Category Population Count, (b) Maneuver Category Percentage

The prediction performance within the 2017 data indicates that the TPOT can be improved with tuning using data that covers a wider variety of maneuver scenarios. This indicates that there may be data diversity issue with the training data. One area for future research is to investigate the performance degradation of a trained model that is extended across multiple years.

5. CONCLUSION AND FUTURE WORK

An exploratory study of using ML models presented in this report demonstrated the feasibility of using AI/ML models to perform data association, maneuver detection, and maneuver classification using computed ground-based

electro-optical observations of RSO's in the GEO orbit regime. Promising preliminary performance metrics can be improved through generalizing training data and further tuning of the AE hidden network. From an operational perspective, these models, once optimized, could be deployed at ground observational sites to run alongside traditional methods to perform near-real time data association of the tracker observations (e.g. angles) and to help identify optical records that are affected by maneuvering RSOs. Such analytics, when run commensurate to operation, can aid the operator in rejecting data that that cause orbit estimation to fail.

TPOT and AE were the two ML models investigated. The AE model was used for data association and the TPOT model was used for multi-class maneuver classification; whereas both the TPOT and AE models were assessed for binary maneuver classification. The AE model for data association achieved better than 90% prediction accuracy. The TPOT model performed well for test data that was from the same year as the training data in both the binary and multi-class maneuver detection applications. The AE model performed well for non-maneuver data but poorly for maneuver data, and this led to the possibility of combining different models to produce better model prediction performance. The results indicate that, in association with traditional statistical filtering techniques, the ML models could provide complimentary techniques to enhance RSO identification, maneuver detection, and maneuver classification for the space awareness communities.

For future work we may address the following topics:

- Training ML models with unbalanced (and more diverse) maneuver data
- Creating multiple model-types to enhance the performance of individual models
- Applying Generative Adversarial Networks (GAN) to produce more comprehensive training data
- Evaluation of actual ground-based tracking data to assess accuracy and timing performance for possible deployment into operational platforms
- Applying these ML models to resolve identification for closely spaced RSOs and thus reduce the number of unknown objects that are closely spaced
- Applying these ML models to filter out outlier observations that cause the orbital estimation processing to fail thereby improving accuracy of the resulting estimated state.

6. ACKNOWLEDGEMENT

We thank our government sponsors for the opportunity to conduct and report on this emerging technology. We also express our gratitude for the assistance of Dr. Brian Pankau and Mr. Patrick Bernard, both of L3Harris, with helpful discussions of the results, as well as for their critical review of the manuscript. Finally, we thank research associate Shiva Iyer of the University of Texas at Austin for his assistance with OrbDetPy.

7. APPENDIX – Definition of the Feature of Training Data

All parameters in the training data are derived from the following input data: optical observation (RA/DEC), location of the ground optical tracker, and orbital state at the epoch of the observation.

The following table and corresponding graphics describe the parameters associated with the training data for ML models:

Parameter Name	Description
RaResidual, DecResidual	Difference between observed and expected Right-Ascension and Declination (expected is from orbital state as viewed by the ground-tracker)
AzObs, ElObs	Observed Azimuth and Elevation angles as converted from oberved RA and DEC
AzRes, ElRes	Difference between observed and expected Azimuth and Elevation (expected is from orbital state as viewed by the ground-tracker)
AlphaObs, BetaObs	Two angles that defines the unit line-of-sight (ULOS) vector from the ground-tracker to observed (RA/DEC) in the Earth-Fixed Frame
AlphaResidual, BetaResidual	Difference between observed and expected Alpha and Beta angles (expected is from orbital state as viewed by the ground-tracker)
deltaULOSeci_X, deltaULOSeci_Y, deltaULOSeci_Z	Difference between the observed and expected unit LOS vector from the ground tracker to the observed and expected position of the RSO in the ECI frame(expected is from orbital state as viewed by the ground-tracker)
ULOSenuObs_X, ULOSenuObs_Y, ULOSenuObs_Z	Observed unit vector line-of-sight from ground tracker to the RSO in the topographical coordinate of East-North-Up (ENU) frame
ULOSECRObs_X, ULOSECRObs_Y, ULOSECRObs_Z	Observed unit vector line-of-sight from ground tracker to the RSO in the Earth-fixed coordinate frame, aka, Earth-Centered-Rotating (ECR)
deltaULOSECR_X, deltaULOSECR_Y, deltaULOSECR_Z	Difference between the observed and expected unit LOS vector from the ground tracker to the observed and expected position of the RSO in the ECR frame (expected is from orbital state as viewed by the ground-tracker)
FovX, FovY, FovZ	components of the unit vector to the observed RSO in the FOV coordinate frame form by line-of-sight vector to the RSO and the zenith vector such that the z-axis is the negative of unit LOS vector, the x-axis is the vector cross-product between the LOS and the zenith unit vector, and the y-axis completes the right-handed triad

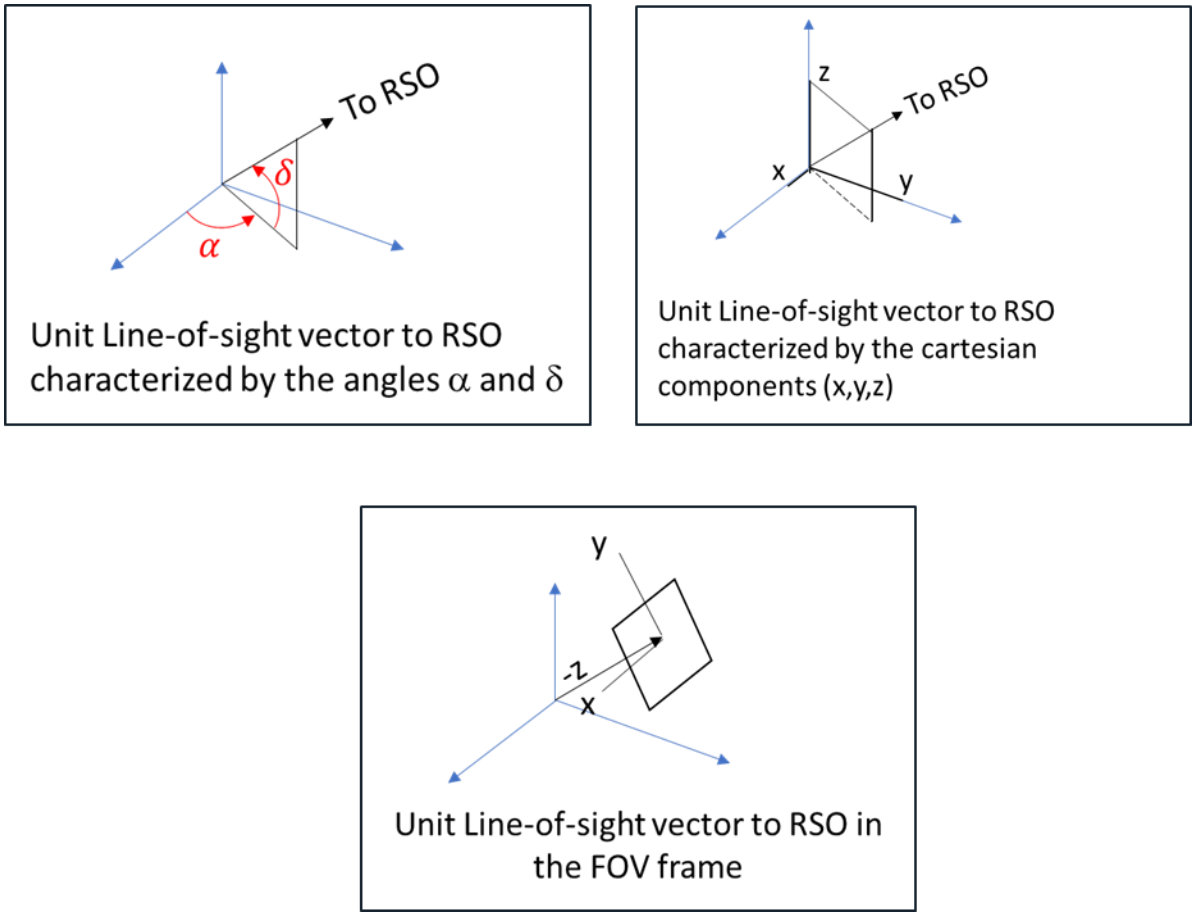


Figure A1 – graphical depiction of derivable angles and components of the unit vector line-of-sight from ground tracker to the RSO (expected and observed) in various coordinate frames

8. REFERENCES

[1] T. Kelecy and M. Jah, "Detection and orbit determination of a satellite executing low thrust maneuvers," *Acta Astronautica*, vol. 66, pp. 798-809, 2010.

[2] M. Holzinger and D. Scheeres, "Object correlation and maneuver detection using optimal control performance metrics," in *AMOS Technologies Conference*, Wailea, HI, September 2010.

[3] T. Kelecy and et al., "Satellite maneuver detection using two-line element," in *AMOS Technologies Conference*, Wailea, HI, September 2007.

[4] D. Lubey and J. Scheeres, "Towards Real-Time Maneuver Detection: Automatic state and dynamics estimation with the adaptive optimal control-based estimator," in *AMOS Technologies Conference*, Wailea, HI, September 2015.

[5] N. Singh, J. T. Horwood and A. B. Poore, "Space object maneuver detection via a joint optimal control and multiple hypothesis tracking approach," in *22nd AAS/AIAA Space Flight Mechanics Meeting*, Charleston, SC, January 2012.

[6] A. Kalur, S. Szklany and J. Crassidis, "Space Object Data Association Using Spatial," *Journal of Astronautical Sciences*, pp. 1708-1734, 2020.

[7] A. Milani, G. Tommei, D. Farnocchia, A. Rossi, T. Schildknecht and R. Jehn, "Correlation and orbit determination of space objects based on sparse optical data," *Monthly Notice of the Royal Astronomical Society*, pp. 2094-2103, 2011.

[8] W. Faber, I. Hussein, J. Kent, S. Bhattacharjee and M. Jah, "Optical data association in a multiple hypothesis framework with maneuvers," in *AAS/AIAA Astrodynamics Specialist Conference*, Stevenson, 2017.

[9] R. Abay and et al., "Maneuver detection of space objects using Generative Adversarial Networks," in *AMOS Technologies Conference*, Wailea, HI, 2018.

[10] R. Linares and R. Furfaro, "Space Objects Maneuvering Detection and Prediction via Inverse Reinforcement Learning," in *AMOS Technologies Conference*, Wailea, HI, 2017.

[11] C. Shabarekh and et al. , "Efficient Object Maneuver Characterization For Space Situational Awareness," in *32nd Space Symposium*, Colorado Springs, CO, April 11-12, 2016.

[12] R. Chalapathy and S. Chawla, "DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY," 24 January 2019. [Online]. Available: https://www.semanticscholar.org/paper/Deep-Learning-for-Anomaly-Detection%3A-A-Survey-Sydney-Centre/a2e667e4382aaa8e02a17d0522c1a910790ab65b?force_isolation=true#citing-papers. [Accessed 23 June 2021].

[13] S. Russo and et al, "Anomaly detection using deep autoencoders for in-situ wastewater systems monitoring data," in *Proceedings of the 10th IWA Symposium on Systems Analysis and Integrated Assessment*, Copenhagen, Denmark, 2019.

[14] S. Chaurasia, S. Goyal and M. Rajput, "Outlier Detection Using Autoencoder Ensembles: A Robust Unsupervised Approach," in *2020 International Conference on Contemporary Computing and Applications (IC3A)*, Abdul Kalam Technical University, Lucknow, India, 2020.

[15] F. Chollet, "The Keras Blog," Keras, 14 May 2016. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html. [Accessed 23 June 2021].

[16] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd and J. H. Moore, "Automating biomedical data science through tree-based pipeline optimization," *Applications of Evolutionary Computation*, pp. 123-137, 2016.

[17] "orbdetpy," UT-Astria, 13 August 2020. [Online]. Available: https://github.com/ut-astria/orbdetpy. [Accessed September 2018].

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[19] R. S. Olson, "TPOT," [Online]. Available: http://epistasislab.github.io/tpot/]. [Accessed 29 July 2020].

[20] J. Hale, "Towards Data Science," Medium, 21 August 2018. [Online]. Available: https://towardsdatascience.com/tpot-automated-machine-learning-in-python-4c063b3e5de9. [Accessed 29 July 2020].

[21] T. Kelecy and M. Jah, "Detection and orbit determination of a satellite executing low thrust maneuvers," *Acta Astronautica*, vol. 66, pp. 798-809, 2010.

[22] D. Tailor and D. Izzo, "Learning the optimal state-feedback via supervised imitation learning," arXiv.1901.0239v2.

[23] M. J. H. Walker, B. Ireland and J. Owens, "A Set of Modified Equinoctial Orbit Elements," *Celestial Mechanics*, vol. 36, no. 4, pp. 409-419, August 1985.

[24] R. Olson, R. Urbanowicz, P. Andrews, N. Lavender, L. Kidd and J. Moore, "Automating biomedical data science through tree-based pipeline optimization," *Applications of Evolutionary Computation*, pp. 123-137, 2016.

[25] A. Pastor, D. Escobar, M. Sanjurjo-Rivo and A. Águeda, "Object Detection Methods for," in *AMOS Technologies Conference*, Maui, HI, 2019.

[26] L. Pirovano and et al, "Probabilistic data association based on intersection of Orbit Sets," in *AMOS Technologies Conference*, Maui, HI, 2018.