Ensemble and Streaming Data Machine Learning Models for Data Association and Maneuver Classification of Resident Space Objects

Triet Tran
Cornerstone Consulting & Services, LLC, MOSSAIC Program

Anthony N. Dills
L3Harris Technologies, Inc., MOSSAIC Program

James Crowley
L3Harris Technologies, Inc., MOSSAIC Program

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 problem of mis-tagging and Uncorrelated Tracks (UCTs) is still a challenge in processing observations today. This problem is in part due to unknown maneuvers that occur between successive revisits of the tracking source on the RSOs. Advanced techniques in statistical filtering have shown 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 address this challenge.

Feasibility of using AI/ML to perform Data Association (DA) and Maneuver Detection/Classification (MD/MC) of the Galaxy-15 in the Geosynchronous (GEO) orbit regime has been demonstrated and reported previously using publicly available Wide Area Augmentation System (WAAS) data. The various ML techniques such as Tree-Based Pipeline Optimization Tool (TPOT) and Autoencoder (AE) were shown to achieve better than 90% accuracy in both data association and maneuver classification [1].

The current work demonstrates improved performance over our previous reported techniques by using the Ensemble ML technique. This technique combines the various ML models into a single model in a way that improves the overall performance over individual models. We show that this technique is applicable to both DA and MD/MC using publicly available data on the Galaxy-15 satellite. Characterizing the 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 astrometric 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. Using open-source template ML models from Keras and Scikit-Learn libraries, we compare the performance of individual models to that of the ensemble model using feature input observational data derived from WAAS data from Galaxy-15 as well as data from a fictitious satellite artificially placed close to the Galaxy-15 location. Our work also addressed the training issue of data staleness that points to the practice of model retraining using the streaming data technique. Preliminary results using ensemble ML technique indicate that for both data association and maneuver classification, the accuracy improved from 88% and 85%, respectively in previous work to at least 94% overall.

In summary, with the encouraging results from the current work, we surmise that there is real potential for improved information gain by combining statistical filtering techniques and ML ensemble techniques that will be of benefit to the space domain community. In our next step, we will consider the enabling capability of building training data from historical TLEs for any catalogued RSOs. This activity is crucial for the success of applying ML techniques to address current challenges, since historical TLEs are abundantly available for most RSOs but precision ephemerides are rare for most RSO's of interest.

1. INTRODUCTION

Data Association (DA) and Maneuver Detection/Classification (MD/MC) still present many challenges to the operations of the Command and Control (C2) centers and to operational tracking systems for Space Domain

Awareness (SDA) [2]. Mis-tagging of observations and Uncorrelated Tracks (UCTs) are two of the many difficulties to overcome to improve catalog maintenance and building [3].

In the past two decades there have been significant contributions in data association and maneuver detection using modern filtering techniques and optimal control algorithms ([4] [5] [6] [7] [8]). More recently, contributions in data association and maneuver detection using ML modeling techniques ([1] [9] [10] [11] [12]) show promising results. In [9] for example, Abay et. al. illustrate that Generative Adversarial Networks can learn the normal behavior of a resident space object under simulated natural forces and detect maneuvers as anomalies, whereas Shabarekh and et al. [11] use an unsupervised Interval Similarity Model to establish correlated patterns of life between astrometric observations and associated maneuvers.

Our recent study of data association and maneuver classification [1] demonstrated further that ML techniques can play a significant role in contributing toward the SDA mission. More specifically, our previous work demonstrated the applicability of ML models to DA and MC using maneuver events of Galaxy-15 derived from the Wide Area Augmentation System (WAAS) data. The models achieved classification accuracy > 90%, a very encouraging result without any tuning being attempted. In this paper we demonstrate the performance of ensemble and streaming ML models in DA and MD, which show significant improvement over our previous models. Briefly, the ensemble ML technique seeks to combine the predictions of individual models into a single, superior prediction, while the streaming ML technique trains and predicts data sequentially in time. Both of the model types considered in this study are appropriate for real-time operational deployment.

The motivation to consider using ML techniques stem from their many successful applications to a variety of problems, such as finance (e.g., credit scoring [13] [14], stock market prediction [15]), medical diagnostics [16] [17], cybersecurity [18], agriculture [19], and others. The immediate next step is to test the ML models of this study with operational data. The combination of modern filtering techniques and ML models may be a worthy approach to achieve a more robust processing system that can improve the accuracy of cataloged orbital states and reduce observation mis-tagging and UCTs.

The organization of the paper is as follows: Section 2 reviews the ML techniques used in the current study, Section 3 explains how training data was generated, Section 4 discusses the result of training and prediction of the models, and Section 5 provides concluding remarks and the identification of future work.

2. MACHINE LEARNING MODELING TECHNIQUES

The focus of this work and resulting application reported in Section 4 cover two different training techniques: (1) the ensemble ML technique and (2) the streaming training technique. The initial design and capabilities of these techniques are discussed.

2.1 Ensemble Modeling Technique

Ensemble learning is a technique of training multiple models and combining their predictions to reach overall performance superior to that of any individual model. The technique is based on the intuition that the wisdom of the crowd in general is better than that of an individual as discussed further in [20] and [21].

There are two key ingredients to the success of the ensemble learning technique: (1) within reasonable limits, an ensemble model's prediction accuracy improves as the number of new models are added to the ensemble until eventually the addition of new models ceases to bring new information; and (2) diversity of the individual ensembled models, which has been shown in [20] to improve overall accuracy.

As the efficacy of the ensemble model is confirmed for data association and maneuver classification in the current study, we will address how to best determine how many and what models to use based on the diversity of the models; even if the models have inherent noise in the predictions, [20] and [21] address how variance reduction is one of the key characteristics of the ensemble technique. We will undertake, in future effort, model tuning to optimize the utility of information from the observation data for decision making.

2.1.1 Ensemble Model Creation

The ensemble ML technique consists of two main aspects: (1) the structure and flow of the training stage and (2) the combination of the individual model predictions [21].

Structure of the training stage

There are different approaches to training the ensemble models; however, we can group them into three main types: Bagging, Stacking, and Boosting ([20] [21] [22]).

Bagging, short for bootstrapping and aggregation, is a technique where a single type of ML model is used, but individual instances of said model are trained using a different set of sample data – each sample dataset being a random selection of equal size taken from a master dataset. The predictions from each trained model are then aggregated, often via a statistical averaging or voting.

Stacking is a technique that uses predictions from multiple types of trained models as inputs to the ensemble model, which subsequently makes the final prediction based on a specified classifier to the inputs.

Boosting is a technique which incorporates new models into the ensemble sequentially, iteratively improving performance over multiple training intervals. During an iteration, model predictions are combined for an overall prediction, which is measured against the known truth of the training dataset to determine the training error. For each iteration, a new model is created and trained with weights based on prior training errors in order to sequentially improve prediction accuracy.

Combination of the individual model predictions

There are three main combination schemes that can be implemented in the ensemble model prediction: (1) Hard Voting, which makes the ultimate classification decision based on a simple majority vote from constituent models; (2) Soft Voting, which makes the classification decisions by summing of probability associated the classifications from each model and choosing the maximum sum; (3) Averaging, which simply takes an average of the prediction from each model. In each of these voting schemes, weights can be included to favor specific models over others. For the purposes of applying the ensemble method to data association and maneuver detection, the combination schemes implemented were hard voting, soft voting, and weighted soft voting.

2.1.2 ML Models Applied to Ensemble Learning

For the purposes of this work, base machine leaning models were sourced from the scikit-learn library [23]. This library contains a selection of supervised learning models, and an analyst can choose as many models as desired to be considered for the ensemble technique.

Additionally, we used the Voting Classifier of the same library to aggregate the results of the existing trained models, each of which is fed the same test dataset. The Voting Classifier provides no post-processing, other than specifications made by the user to have final predictions assessed with either hard or soft voting and any weights to apply to sub-models.

For both the data association and maneuver classification problems investigated in the current study, we limit the selection of models used in the ensemble to the ones listed below:

a) Logistic regression is a linear model for classification. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function. This implementation can fit binary, One-vs-Rest, or multinomial logistic regression with optional, or Elastic-Net regularization.

b) RandomForest classifier. A random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the prediction accuracy and to control over-fitting.

c) AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

d) Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

e) Multi-layer Perceptron classifier is one of the earliest and simplest types of neural network, where the architecture consists of multiple layers each with a specified number of neurons with attribute of bias and weight where connections are made from all neurons of one layer to all neurons of the adjacent layer. Training is achieved by what is known as backpropagation to drive the output error toward zero in a supervised learning mode. This model optimizes the log-loss function using Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm or stochastic gradient descent.

2.2 Streaming Training Technique

Stream or on-line training techniques, when applied to ML model training, allows the model to update every time it is presented new data, either as a single record or as a batch. The most common Python packages include `creme` and `scikit-multiflow`, which were merged to provide a leading method called `riverML`. [24] The stream analysis supporting here in was performed using `riverML` and its two methods (*learn_one* and *learn_many*) to train and learn on streaming data. These methods allow the model to learn (and predict) using one record at a time or a small batch of records, respectively and support ubiquitous operational scenarios for data flow

For the current analysis, we compared Logistic Regression (LR), Passive Aggressive (PA) [25], and perceptron classifiers. All 3 implementations in `riverML` support *learn_one* whereas, PA is the only one that does not support *learn_many* (as at the time of this publication). Both methods of learning for each of the models are compared in the analysis discussion in Section 4.

3. TRAINING DATA GENERATION METHODLOGY

In our previous and current study, our observables are optical data of Right Ascension (RA) and Declination (DEC) as observed from a specified ground tracker. To capture the orbit of an RSO at GEO, we use the high-precision ephemeris for Galaxy-15, known as Wide Area Augmentation System (WAAS) data, which is available publicly from the Federal Aviation Administration (FAA). From this data, feature vectors used to train the five ML models listed in Section 2.1.2 were generated, with separate data generated and training performed for the maneuver classification and data association problems. Ultimately, more than 310,000 records were generated using this training record generation technique, spanning over 3 years of Galaxy-15 WAAS data for 2016, 2017, and 2018 for MC and double the number of records for DA. These were divided 75/25 into training and cross-validation data, respectively, with the 75% used to train the individual models listed in Section 2.1.1 prior to incorporation into the ensemble model and the 25% reserved as test data.

3.1 Generation of Feature Vectors for Maneuver Classification

The technique used in this study to capture maneuver attributes in the WAAS dataset was implemented in previous work [1], and each ephemeris record was labeled accordingly to recognize station-keeping maneuvers in the East-West (EW), North-South (NS), and Radial (Rad) directions in support of ML model training. The full explanation of the maneuver detection methodology applied to the WAAS data is included in Appendix A.

The training data used in this work is derived from the residuals between simulated optical observations (which may or may not include the effect of maneuver) and the expected optical observations, which assume no maneuver is executed. Individual simulated observations were generated through transformation of the WAAS high-precision state vector into the observation space at a given epoch, with the corresponding expected observation generated via

the unscented transform sampling technique described below. This limits the training data to only included measurement information that an observer would have in-situ. A further description of the observation generation process can be found in Appendix B.

From each simulated and expected observation pair, a feature vector was generated, consisting of 28 features based on the measurement geometry. These features include the residuals in the measurement and the line-of-site vector (and its residuals) between the ground station and target transformed into various reference frames. The full details of each of the 28 features is discussed in Appendix A.

Feature vector datasets and ML models trained with them are unique to each ground station/satellite pair, and application of this ensemble technique to other ground station/satellite pairs would necessitate the generation of new feature vector datasets and retraining of models based on them. Although this may seem like a limitation, because of the uniqueness of the data due to viewing geometry, it is necessary to build training data accordingly and data generation can easily be performed with today's computing resources.

3.2 Unscented Transform Sampling

Since our training data requires expected observations based on catalogued states to be generated, we employed a technique called Unscented Transform (UT) sampling to ensure statistical coverage of the effect of state uncertainty in the training data. For each initial state taken 24 hours prior to a maneuver, 13 UT states are generated. This set of 13 states are propagated to the end of the observation window, which is 12 hours after the maneuver epoch, and used in calculation of the expected observation at any epoch in that timeframe. Ideally, Monte Carlo sampling can be used, but it is too expensive computationally to be practical. The UT was first investigated by Julier and his colleague in association with the Unscented Transform Kalman Filter [26]. More recently, it was demonstrated in [27] how the UT sampling can represent the Probability Distribution Function (pdf) of a population and any non-linear transformation therefrom. See Appendix C for more detailed discussion of the UT sampling technique.

3.3 Generation of Feature Vectors for Data Association Training via Shadow Ephemeris

To test the effectiveness of the ensemble technique on the data association problem, a secondary target that is in a similar-enough orbit to be mis-tagged as the primary is required. For this reason, we have created a shadow ephemeris for a fictitious RSO that is placed optically close to the real Galaxy-15 RSO. For this study, the optical separation between Galaxy-15 and the Galaxy-15 Shadow RSO is about 8 arcseconds. The shadow ephemeris was generated based on the WAAS ephemeris data but offset in position from the actual Galaxy15 position in the East-West direction. Further details of how this shadow ephemeris was created are given in Appendix D. Feature vector training and test data generation was performed using the Galaxy-15 and Galaxy-15 shadow observations, again implementing the technique described in Appendix A.

4. RESULT AND DISCUSSION

4.1 Data Association using Ensemble Technique

We applied the ensemble model to the problem of observation data association of targets in neighboring orbital regimes. Binary classification was performed, using the Galaxy15 as one label and the emulated Shadow Galaxy15 as the other.

We present the performance of the ensemble ML model using the confusion matrices, showing the accuracy of data association for a hard, soft, and weighted voting techniques in Fig. 1. For weighted voting, we assigned the following weighs after observing the performance of individual models: Logistic Regression (1), Random Forest (3), Ada Boost (2), Bagging Classifier (3), and Multilayer Perceptron (2).

Overall, the ensemble model has greater than 96% accuracy in either classification category for all three voting techniques, with the soft voting accuracy being higher than that of hard voting, and weighted voting having the best results of the three. For comparison, the performance of the single ML model in our previous work [1] achieves 98% accuracy for correctly identifying observation records belonging to Galaxy15 and 88% accuracy for those that do

not. The results shown here demonstrate that all voting modes of the ensemble ML model outperform the single ML model.

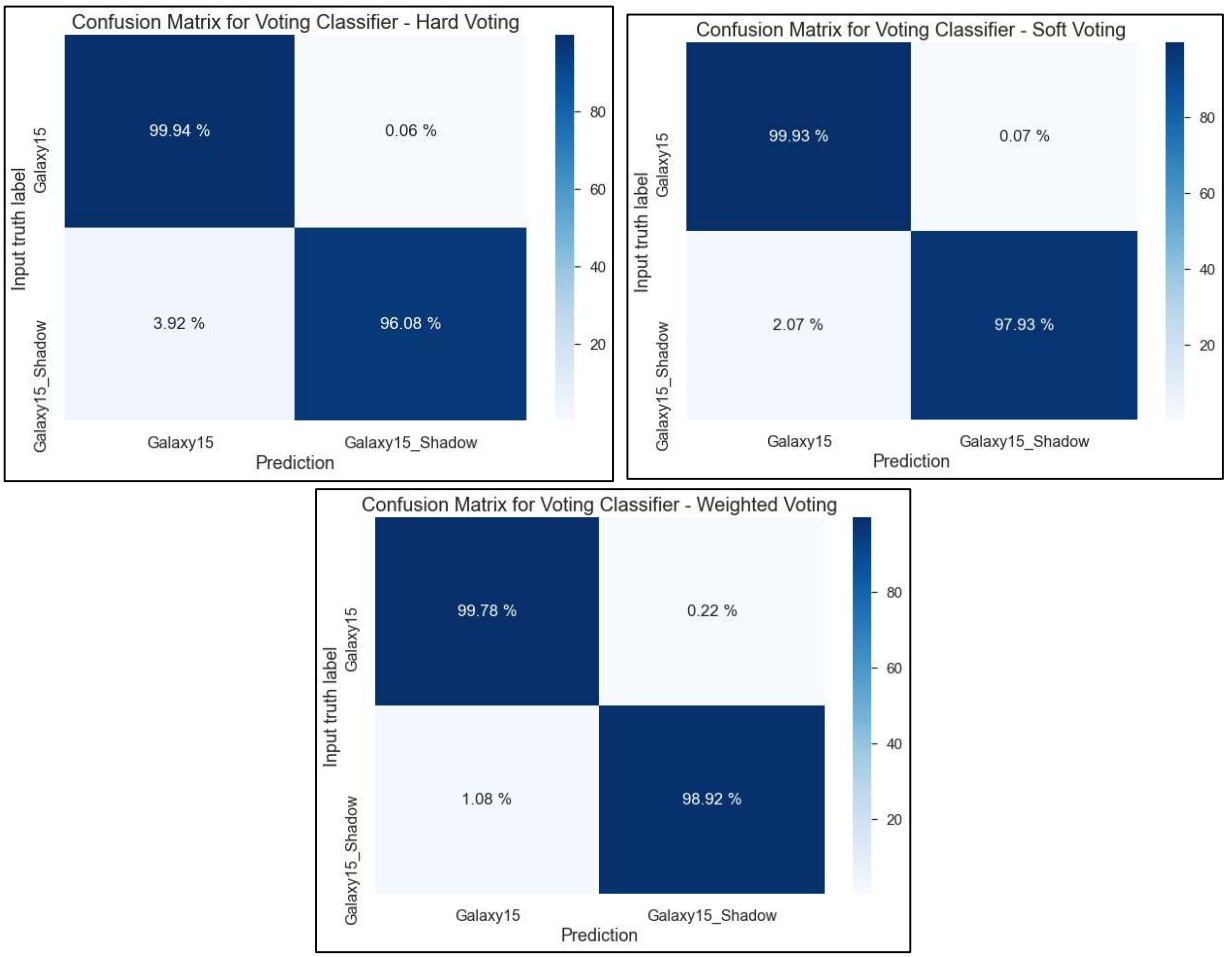

Fig. 1. Confusion matrices for Data Association: top-left: Hard voting, top right: Soft Voting, lower center: Weighted Voting

4.2 Maneuver Classification using Ensemble ML Technique

The ensemble model was also applied to the problem of maneuver classification. In data processing, feature vector was labeled as displaying either no maneuver, or a maneuver in the EW, NS, or Rad direction. For this test, all types of station keeping maneuver were grouped into a single category, resulting in the ensemble model providing binary classification between two categories: Maneuver (1) and Non-Maneuver (0).

We present the performance of maneuver classification using the confusion matrices, showing the accuracy under the three voting techniques in Fig. 2. For this test, the same model weights described in Section 4.1 were again implemented for weighted voting. The weighted voting technique resulted in the best performance, followed by soft voting. The ensemble model achieved higher than 92% accuracy in each classification category when implementing weighted and soft voting.

For comparison, the performance of the single ML model in our previous work [1] has accuracy of 85% for non-maneuver and 97% for maneuver, which is very similar to that of the hard voting case of the current study, while the soft-voting and weighted voting modes outperform the single ML model.

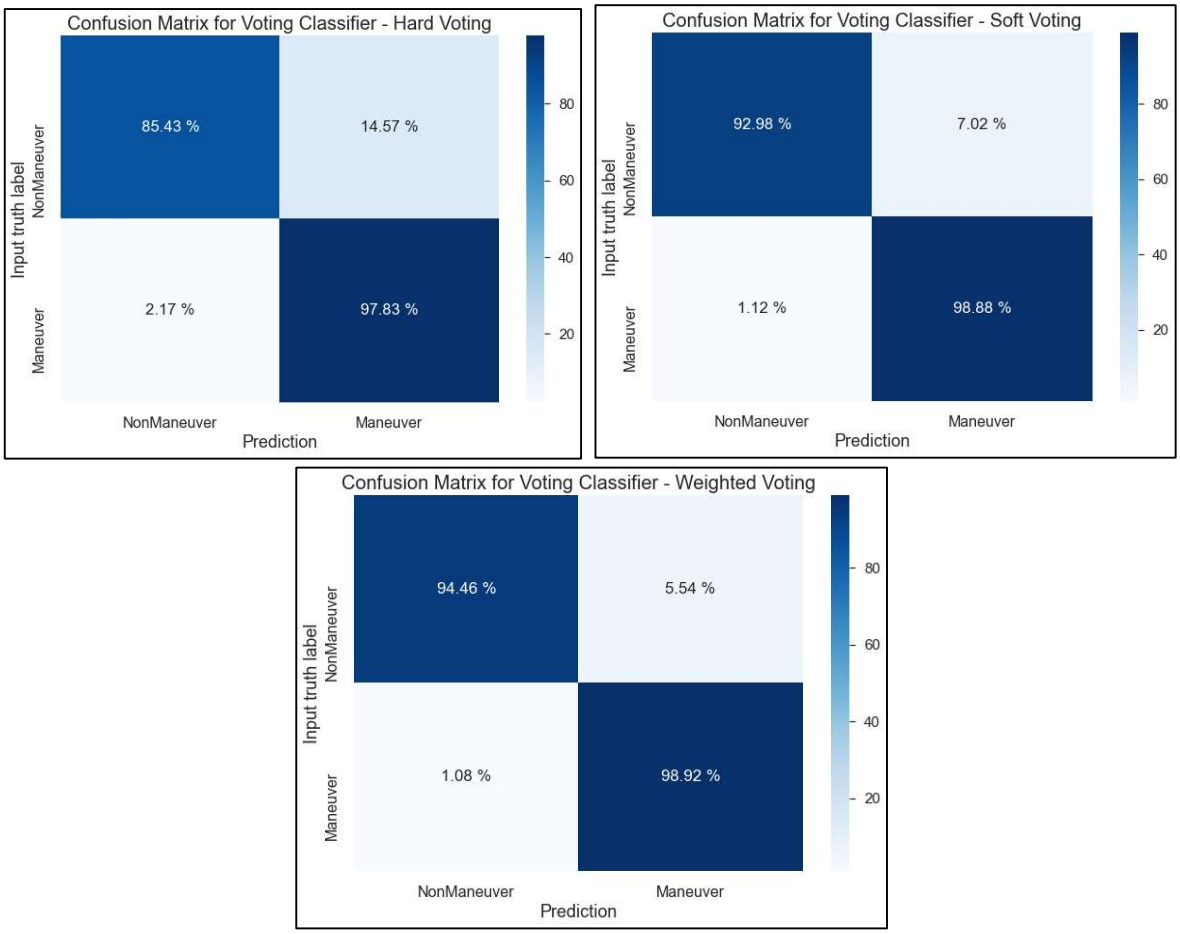

Fig. 2. Confusion matrices for Maneuver Classification: top-left: Hard voting, top right: Soft Voting, lower center: Weighted Voting

4.4 Maneuver Classification using Streaming Data Technique

We compared LR, PA, and perceptron classifiers for both *learn_one* and *learn_many* (except PA) using the same input date as described in Section 3 except as a streamed source and for binary maneuver classification. In general, the PA and perceptron classifiers yielded higher F1 and Receiver Operating Characteristic Area Under the Curve (ROC AUC) metrics than the LR classifier. The single record training scored better than batch-based streaming.

Using the LR classifier, trained one record at a time, the ROC AUC quickly goes to near unity within the first few records; however, as time continues, or as more records are used to train the model, the performance degrades over time as shown in Fig. *3*. Specifically, the ROC AUC quickly achieves a value near one but then begins to slowly degrade. The degradation is not significant but the fact that it degrades is indicative of slightly lower quality of classification prediction. The AUC measures the quality of the model's predictions, so a closer look at the trade of false positives and false negatives may be warranted. This figure, like the ones following this sub-section, are contained the statistical performance at each prediction step (*predict_one* or *predict_many*), captured as ROC AUC, F1, and R^2 (or R2) (on the left ordinate) and Mean Absolute Error (MAE) and Mean Squared Error (MSE) on the right axis. Goodness for ROC AUC, F1, and R2 is closer to unity and a lower MSE and MAE is better.

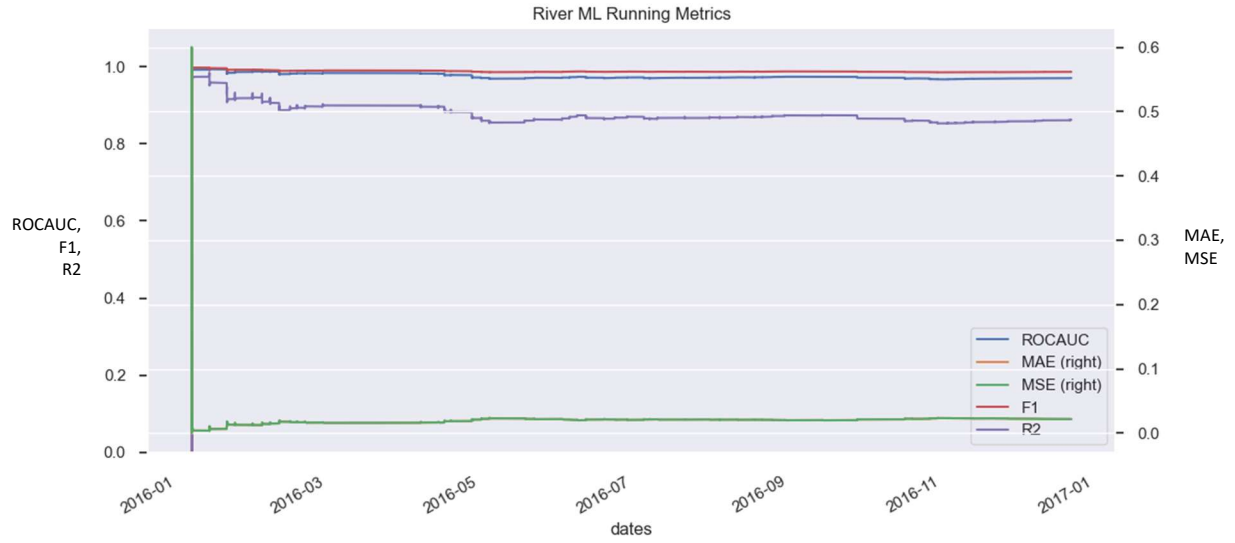

Fig. 3. LR binary maneuver performance during learn_one training

Using a PA Classifier (PAC), trained one record at a time, Fig. *4* shows the ROCAUC and other metrics quickly go and sustain values near unity within the first few records. Similarly, the MAE and MSE performances achieve near perfection at the same rate.

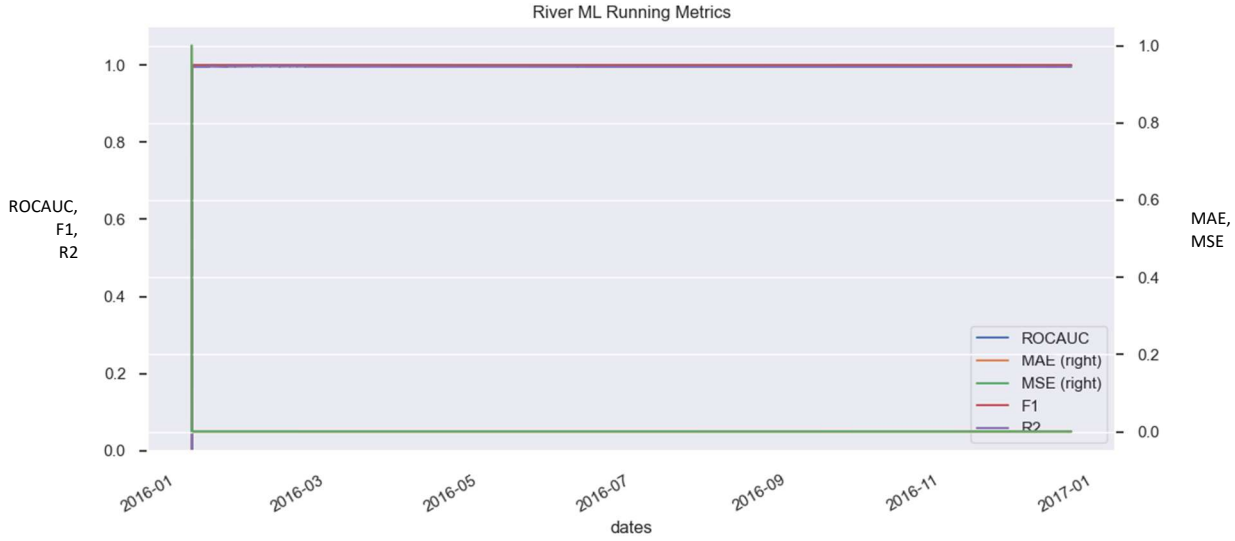

Fig. 4. PAC binary maneuver performance during learn_one training

The perceptron classifier with single record learning performed nearly identical to the PAC as shown in Fig. *5*.

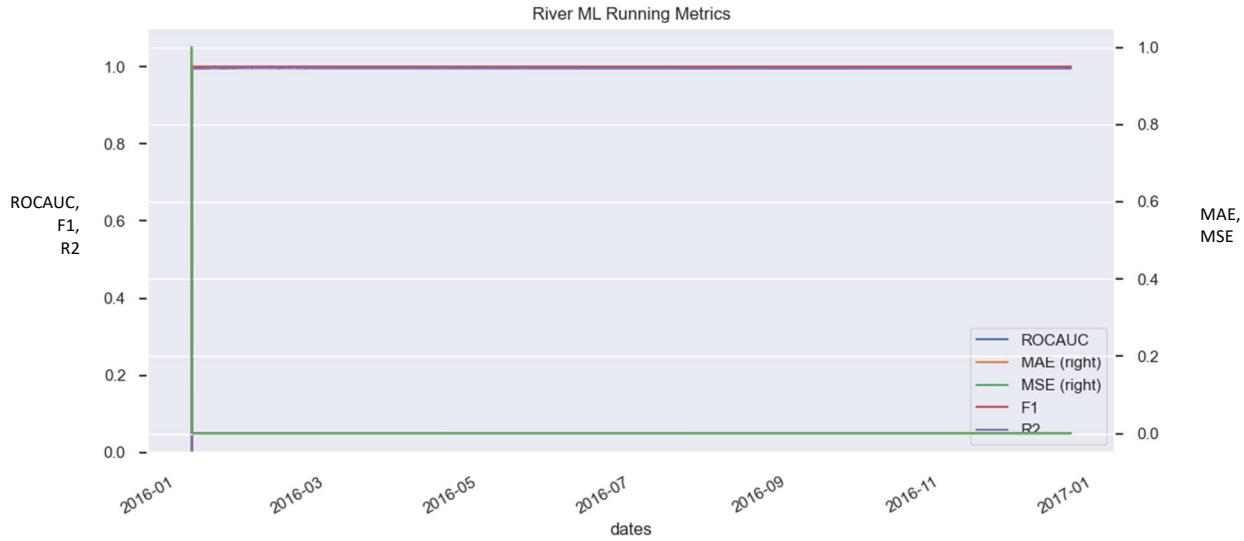

Fig. 5. Perceptron classifier binary maneuver performance during learn_one training

These performances demonstrate the potential to achieve high accuracy in maneuver binary classification when examined during a single record training environment. There are, however, operational use cases where data present themselves as chunks or batches. In this case, it is important that we report on batch performance using the `riverML`'s *learn_many* and *predict_many* functions. Of the 3 models under investigation, only LR and perceptron classifiers support batch learning.

The batch scenario chosen was a daily batch to ensure an operational use case where input features require pre-processing throughout the day. As such, the 2016 evaluation data contained 93 unique days, with each day typically containing between 300 and 2000 individual records.

The next two figures (Fig. *6* and Fig. *7*) depict the training results of batch processing for the LR and perceptron classifiers. Recall that the `riverML` PAC does not currently support batch processing. Contrary to the single record learning, batch learning clearly showed a performance difference between the LR and perceptron models, with the LR the clear superior. Unlike the single line learning, the batch learning statistics exhibit realism in that it takes at least a few days to achieve good performance (rather than only 1-2 records). At this time, the authors have not investigated this behavior but recognize that the models do need further tuning

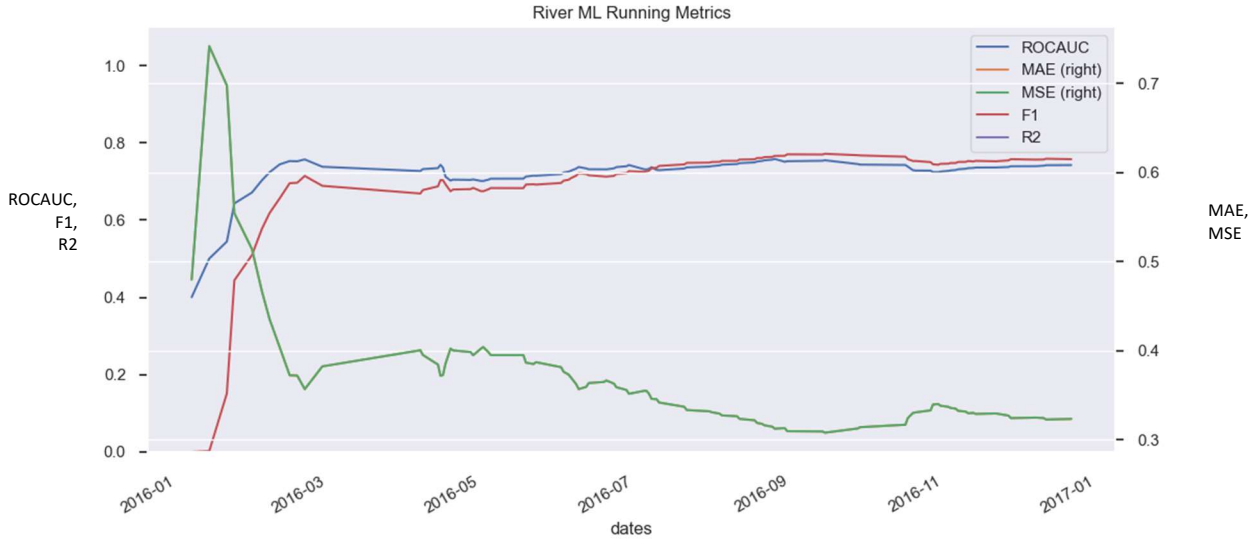

Fig. 6. LR binary maneuver performance during learn_many training

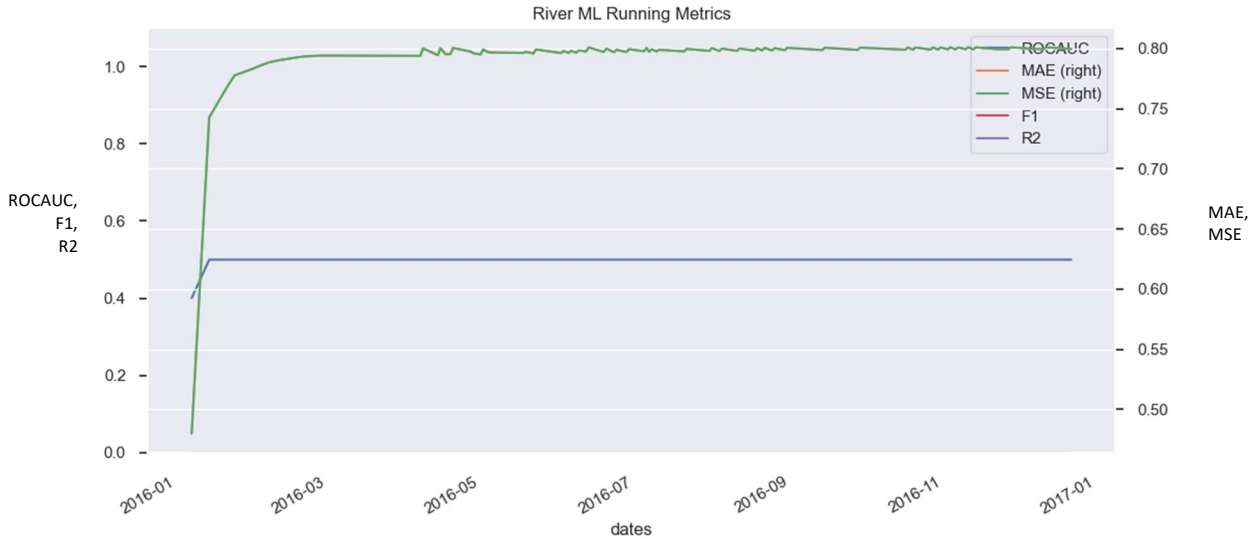

Fig. 7. Perceptron binary maneuver performance during learn_many training

5. CONCLUSION AND FUTURE WORK

In our continued effort to address the DA and MC problems using ML modeling techniques, we have shown that these techniques can provide substantial benefits to SDA mission, specifically at addressing mis-tagging and UCTs.

We have described the ML techniques applied in this work, the ensemble ML technique and the ML streaming training technique, as well as the data involved in the training of the ML models applied in this work. The performance of the ensemble learning affirms the efficacy of this ML technique in many other areas of application. The key success of the application lies in constructing the proper training data with the methodology of taking advantage of the unscented transform sampling technique (in place of using the exhaustive Monte Carlo sampling) to characterize the orbital uncertainties of the catalog state.

In summary, the ensemble ML models achieve better than 98% accuracy for both data association and maneuver classification while the streaming ML technique varies considerably between different models with classification

performance exceeding ranging from 75% to 98% F1 scores. It is too early in the investigation to state that streaming training models perform better than the ensemble techniques.

For future work, we will explore the applicability of generating training data from the abundance of the Two-Line Elements (TLEs) for many RSOs. In addition, we will pursue using operational observations and catalog data to demonstrate the viability of these ML techniques for operational deployment. Furthermore, we will work toward the feasibility of model deployment for real-time operations and compare the results to current operational capabilities. In addition, model tuning will be undertaken to optimize performance of the models in the current study. Finally, we will consider a longer-term goal of combining modern filtering techniques with ML techniques and believe this approach will combine the best of both worlds toward achieving further benefits for the SDA mission.

6. ACKNOWLEDGEMENT

We thank our government sponsors for the opportunity to conduct and report on this emerging technology as part of MOSSAIC contract FA8823-20-C-0004. 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.

7. REFERENCES

[1] T. Tran and A. Dills, "Toward Data Association and Maneuver Classification using Machine Learning Techniques for GEO RSO's," in *AMOS Tech Conference*, Maui, HI, 2021.

[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] G. Der, Astrodynamics Algorithm for Rapid Space Catalog Building in Matlab, Colorado Springs Dallas Los Angeles: DerAstrodynamics.com, ISBN 978-1-69344-601-8, 2021.

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

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

[6] 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.

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

[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] J. Decoto and D. Dayton, "arXiv:2001.05855v1 [cs.CV]," 9 Janurary 2020. [Online]. Available: https://arxiv.org/pdf/2001.01259?force_isolation=true. [Accessed 27 May 2022].

[13] S. Netra, S. Dahiya and S. Handa, "Credit scoring using Ensemble of various classifiers on reduced feature set," *Industrija,* vol. 43, no. No. 4, pp. 163-174, 2015.

[14] S. Parvin and B. Saleena, "An Ensemble Classifier Model to Predict Credit Scoring - Comparitive Analysis," in *2020 IEEE International Symposium on Smart Electronic Systems*, 2020.

[15] S. Mehta, P. Rana, S. Singh, P. Sharma and P. Agarwal, "Ensemble Learning Approach for Enhanced Stock Prediction," in *2019 Twelfth International Conference on Contemporary Computing*, 2019.

[16] A. Sharma and R. Rinkle, "Ensembled machine learning framework for drug sensitiviy prediction," *The Institution of Engineering and Technology,* vol. 14, no. 1, pp. 39-46, 2019.

[17] I. Odeh, M. Alkasassbeh and M. Alauthman, "Diabetic Retinopathy Detection using Ensemble Machine Learning," https://arxiv.org/abs/2106.12545v1?force_isolation=true, 2021.

[18] X. Gao, C. Schan, Z. Niu and Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection," 2019.

[19] A. Khatri, S. Agrawal and J. Chatterjee, "Wheat Seed Classification: Utilizing Ensemble Machine Learning Approach," *Scientific Programming,* vol. 2022, no. 1, February 2022.

[20] C. Zhang and Y. (. Ma, Ensemble Machine Learning, New York: Springer, 2012.

[21] A. Geron, Hands-on Machine Learning wit Scikit-Learn, Keras & TensorFlow, Beijing Boston Tokyo: O'Reilly Meida Inc., 2019.

[22] J. Brownlee, "A Gentle Introduction to Ensemble Learning Algorithms," Machine Learning Mastery, 27 April 2021. [Online]. Available: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/. [Accessed 27 May 2022].

[23] s. learb, "scikit learn 1.11.Ensemble Methods," [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html. [Accessed 27 May 2022].

[24] J. Montiel et al., "River: Machine Learning for Streaming Data in Python," *ArXiv:2012.04740 [Cs],* 2020.

[25] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz and Y. Singer, "Oneline Passive-Agressive Algorithms," *Journal of Machine Learning Research,* vol. 7, no. 19, pp. 551-585, 2006.

[26] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE,* vol. 92, no. 3, pp. 401-421, March 2004.

[27] L. Angrisani, M. D'Apuzzo and R. Moriello, "Unscented transform: a powerful tool for measurement uncertainty evaluation," *IEEE Transactions on Instrumentation and Measurement,* vol. 55, no. 3, pp. 737-743, June 2006.

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

[29] 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].

8. APPENDICES

Appendix A – Feature of the 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.

Table A-1 and corresponding graphics (see Fig. A-1) describe the parameters associated with the training data for ML models:

Table A-1. Features for training data

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

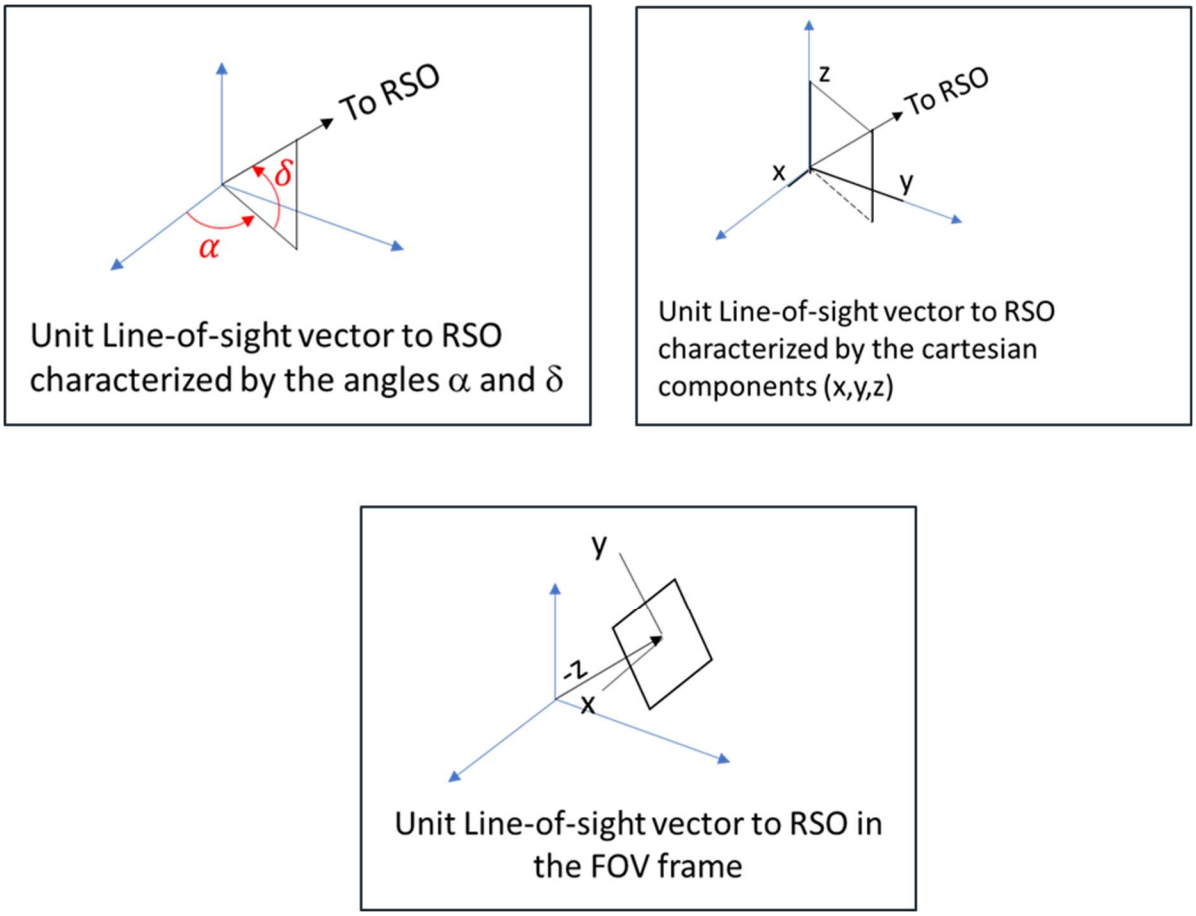

Fig. A-1. Graphical depiction of derivable angles and components of the unit vector line-of-sight from ground tracker to the RSO (expected and observed) in various coordinate frames

Appendix B – Maneuver Detection using High-Precision Ephemeris

With the precision ephemerides such as WAAS data, a simple technique was developed to label the maneuver events automatically. Fig. B-1 illustrates the maneuver detection technique using consecutive records within the ephemeris data. The open source OrbDetPy [28] 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 GEO orbit.

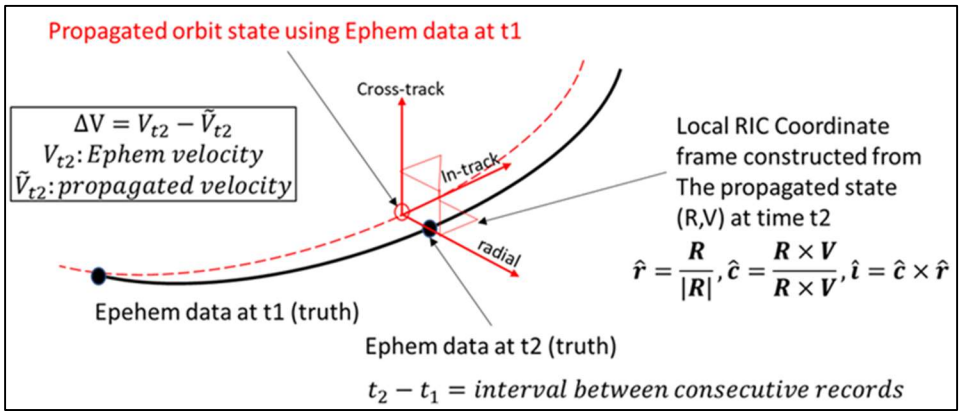

Fig. B-1. Maneuver Detection using precision ephemerides

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 simulated 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.

Appendix C – Unscented Transform Sampling for Orbit Uncertainty

Due to errors inherent in the orbital knowledge of the RSOs, it is necessary to include the effect of orbital errors in the orbit state in the training data. In some cases when the orbital uncertainty is large, it may mask the effect of the maneuver and causes the ML model to be "confused" in the application of the ML technique to maneuver classification. For this reason, we have employed the UT sampling technique to capture the orbital uncertainty in building the training data for the ML models.

The efficacy of using the UT sampling was demonstrated in [27] where the authors used the UT sigma points constructed from the mean and covariance of a state and performed a non-linear transformation of those sigma points. In addition, the Monte Carlo sampling were used to compare against the distribution of the Sigma Points from the UT sampling. The excerpted figures from the Reference as shown in Fig. C-1 show the comparison between the Monte Carlo vs the UT sampling of the electrical impedance before and after going through a non-linear transformation.

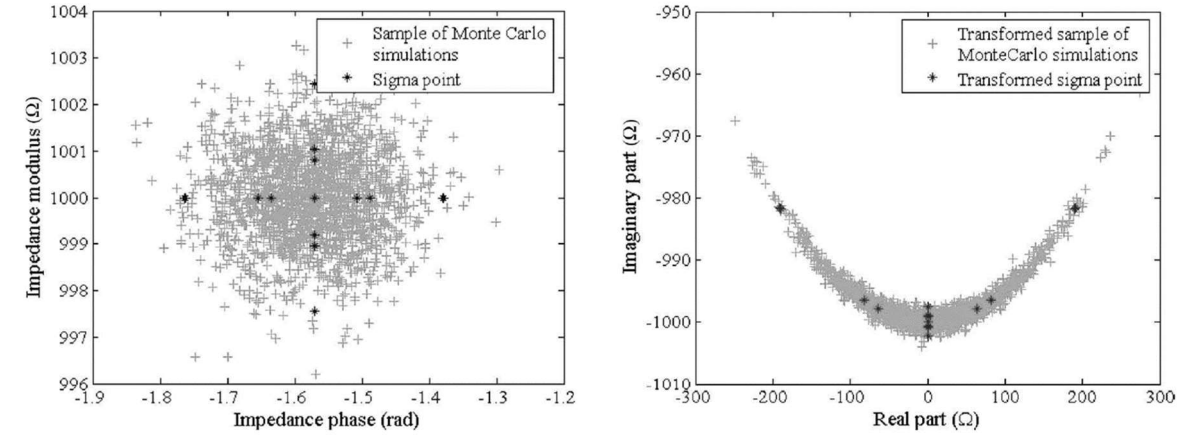

Fig. C-1. Comparison between Monte Carlo and UT sampling - left: initial normal distribution; right: after non-linear transformation

Our internal investigation also showed how UT sampling of the orbital uncertainties helps improve the performance of ML models over non-UT sampling. Motivated by the result of the study in [27], we have employed this technique to sample the pdf of the orbit state uncertainty in our current study.

Appendix D – Generation of the Shadow RSO

Data Association is actually a binary classification problem where, for ensemble ML technique using scikit-learn library, it is necessary to have label data available for training. To this end we created a shadow RSO using the following procedure: for each maneuver event, emulated optical observations are collected 3 hours prior until 12 hours after the event. The sampling frequency is once every 10 minutes. The initial orbit state is taken at 24 hours prior to the maneuver. From this initial orbit state, an offset of 8.7 km is added to the x-component of the position vector. The original state and offset state are propagated to the end of the observation period (12 hours post maneuver event). At each sampling time, the difference in position vector is obtained (Fig. D-1) and then added to the original WAAS record and taken as the ephemeris data for the Shadow RSO (Fig. D-2). This procedure is illustrated in Fig. D-1.

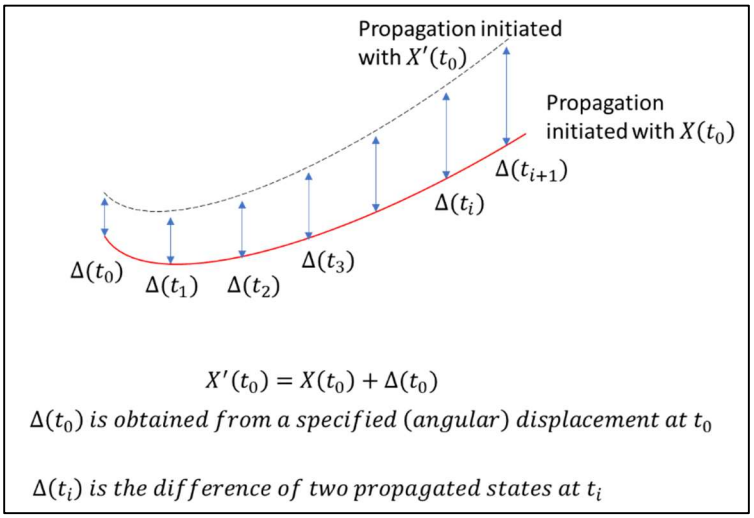

Fig. D-1. Illustration of the generation of the Shadow Ephemeris for a fictitious RSO based on the original WAAS Ephemeris Data for Galaxy-15

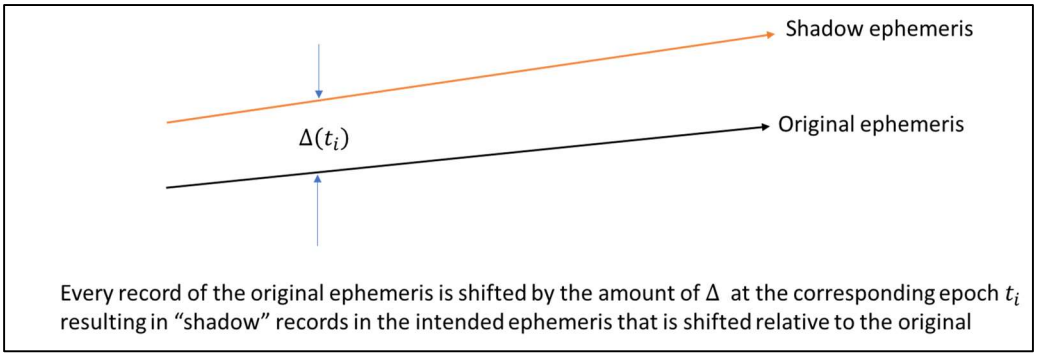

Fig. D-2. Construction of the shadow ephemeris from the original WAAS ephemeris

This procedure allows the shadow ephemeris to capture the same maneuver characteristics of the original ephemeris. The shadow ephemeris is labeled as "other RSO" for the purpose of training the ML model to recognize whether an optical observation belongs to the Galaxy-15 RSO.