USING MACHINE LEARNING FOR ADVANCED ANOMALY DETECTION AND CLASSIFICATION

Ben Lane
ExoAnalytic Solutions
Mark Poole
ExoAnalytic Solutions
Matt Camp
ExoAnalytic Solutions
Jeremy Murray-Krezan
Air Force Research Laboratory

1. ABSTRACT

Machine Learning (ML) techniques have successfully been used in a wide variety of applications to automatically detect and potentially classify changes in activity, or a series of activities by utilizing large amounts data, sometimes even seemingly-unrelated data. The amount of data being collected, processed, and stored in the Space Situational Awareness (SSA) domain has grown at an exponential rate and is now better suited for ML. This paper describes development of advanced algorithms to deliver significant improvements in characterization of deep space objects and indication and warning (I&W) using a global network of telescopes that are collecting photometric data on a multitude of space-based objects. The Phase II Air Force Research Laboratory (AFRL) Small Business Innovative Research (SBIR) project Autonomous Characterization Algorithms for Change Detection and Characterization (ACDC), contracted to ExoAnalytic Solutions Inc. is providing the ability to detect and identify photometric signature changes due to potential space object changes (e.g. stability, tumble rate, aspect ratio), and correlate observed changes to potential behavioral changes using a variety of techniques, including supervised learning. Furthermore, these algorithms run in real-time on data being collected and processed by the ExoAnalytic Space Operations Center (EspOC), providing timely alerts and warnings while dynamically creating collection requirements to the EspOC for the algorithms that generate higher fidelity I&W.

This paper will discuss the recently implemented ACDC algorithms, including the general design approach and results to date. The usage of supervised algorithms, such as Support Vector Machines, Neural Networks, k-Nearest Neighbors, etc., and unsupervised algorithms, for example k-means, Principle Component Analysis, Hierarchical Clustering, etc., and the implementations of these algorithms is explored. Results of applying these algorithms to EspOC data both in an off-line "pattern of life" analysis as well as using the algorithms on-line in real-time, meaning as data is collected, will be presented. Finally, future work in applying ML for SSA will be discussed.

2. INTRODUCTION

In previously published work [1] the Autonomous Characterization Algorithms for Change Detection and Characterization (ACDC) framework for space object characterization with real-time commercially available data was described, see Figure 1. Although there have been diverse prior efforts to build photometric characterization algorithms [2], there have been very few demonstrations of an end to end photometry-algorithm processing chain that is fully integrated with sensors for follow-up tasking; this is the relatively unique contribution of the ACDC capabilities. As shown in Figure 1, a set of autonomous algorithms and application manager will reside inside the Advanced Research, Collaboration, and Application Development Environment (ARCADE) hosted by the AFRL. ARCADE was established in 2012 as a centralized test-bed for all research and development activities related to new Joint Space Operations Center (JSpOC) Mission System (JMS) applications: algorithm development, data source exposure, service orchestration, and software services. In addition to providing JMS with a modern test-bed environment, ARCADE is exploring new business processes for how the U.S. acquires and upgrades its operational information systems [3]. Included amongst these aims are the use of non-traditional data, typically defined as non-Space Surveillance Network, and more specifically in this case commercial-SSA sensing data. Furthermore, AFRL funded ExoAnalytic to develop space object characterization and change detection algorithms with the intent to mature this

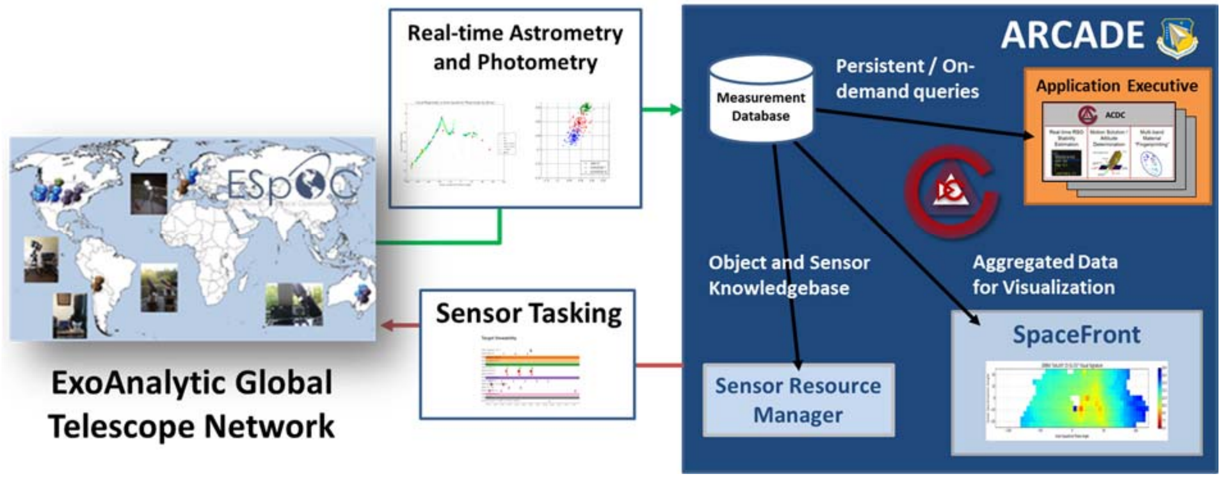

Figure 1 Automated Characterization and Correlation Process

technology through the ARCADE-to-JMS transition process. With a transition process in place for developing and maturing the technology, a sufficient data source is needed to determine the robustness of algorithms also under development in the ARCADE. While supporting the ARCADE goals, the ExoAnalytic Global Telescope Network (EGTN) can provide data for algorithm maturation.

The EGTN collects a continuously increasing amount of photometric data that is available for data mining and machine learning applications. Having full longitudinal coverage, the EGTN spans the geostationary equatorial orbit, the region around which is typically called the GEO belt (see Figure 2). With over 80 telescopes in multiple countries, the network has been operating since 2012 and is currently averaging 100,000 correlated observations per night. The nightly operation for every sensor in the network is customizable and dynamically allocated. Sensors can be dedicated to broad tasking objectives such as GEO search, full-sky search, and client-focused revisit. This mix of sensor operations provides a robust and highly adaptable opportunity to obtain ample data on all a prior expected GEO events. With this sensor architecture, the ETGN can collect photometric data on high interest and new launch GEO objects without any specific feedback loop. Although the EGTN has coverage of other orbit regions, the emphasis of ACDC's automated characterization and state change detection software architecture is primarily employed for objects near the GEO belt.

Currently work is focused on researching the application of various ML techniques for autonomous characterization in six areas. These six goals include i) satellite identification for night to night correlation; ii) satellite bus identification; iii) normal versus anomalous behavior; iv) satellite mission; v) object stability; and vi) satellite-bus and solar panel pointing. This paper will discuss our initial focus on satellite and bus identification as these allow for the most accessible labeled datasets to begin research, while also presenting advanced techniques for space object stability and for satellite bus and solar panel pointing. Although research is ongoing, this paper documents the investigations and results completed up to date. Multiple machine learning algorithms were initially explored, including Principle Component Analysis (PCA) and Hierarchical clustering, and four different supervised machine learning algorithms were implemented and tested for satellite identification: support vector machines (SVM), random forest (RF), k-Nearest Neighbor (k-NN), and stochastic gradient descent (SGD). Two additional algorithms were investigated when classifying the bus type: Perceptron and Multi-Layer Perceptron. For classification of satellites, the accuracy of results relied heavily on the classes of satellites under investigation: satellites expected based on prior knowledge to be in an active state yielded the most accurate and reliable results. Initial results for classifying bus type show promise, with additional investigation required to incorporate other satellite buses for training purposes, and adding more features to the predictive model.

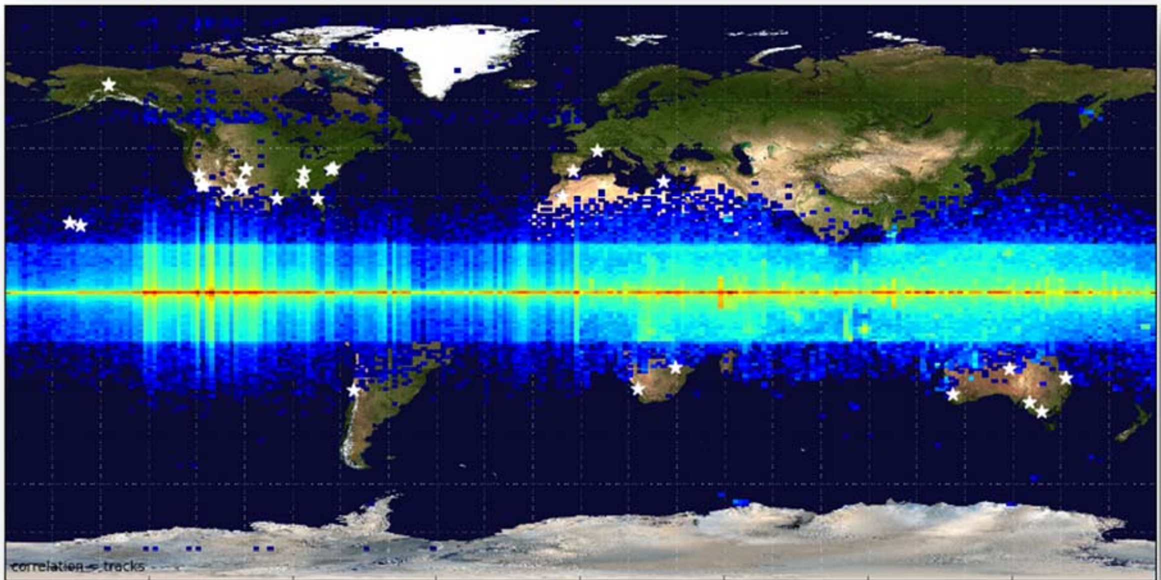

Figure 2 EGTN and coverage of the GEO Belt

3. APPROACH

Generating over 100,000 nightly observations allows for datasets that can often exceed over one million observations for the time period of interest. Still much of this work was conducted on a typical desktop machine. One correlated observation is comprised of fifty data fields extracted from a single raw image taken by telescopes in the EGTN. The observation's NORAD ID, or catalog ID number, is stored as one of these fifty fields (Table 1) and is the output goal for the machine learning algorithm when performing satellite classification. Additional fields such as satellite bus type can be generated using open source data cross-referenced by catalog ID. This additional field is the output for bus classification results.

Table 1: Sample photometric data

sensor_id	Catalog _id	jd	site_lat	site_lon	site_alt	los_az	los_el	los_ra	los_dec	...	solar_dec_ang	geo lat	geo lon
1099	28884	2457258.10	21.92	-159.51	219.0	126.76	50.62	60.64	-3.68	...	11.28	0.02000	227.00
1092	28884	2457168.75	37.07	-119.41	1406.0	201.86	44.67	201.61	-5.81	...	16.65	-0.02000	226.98
12001	28884	2457405.73	31.94	-110.25	1092.0	218.44	45.45	63.85	-5.15	...	-23.48	0.02000	227.00
1013	28884	2457066.73	33.66	-117.64	244.0	206.37	47.54	92.61	-5.40	...	-16.60	0.00162	227.00
1134	28884	2457450.90	21.92	-159.51	219.0	126.83	50.59	178.62	-3.60	...	-10.07	-0.02000	226.99

Due to the popularity of machine learning techniques in recent years, many programming environments have relevant and mature code libraries. Two popular languages are R and Python[1]. R started as a statistical computing language and has a very large number of community generated packages. R was experimented with early in the research, but our investigations were hampered by memory constraints that limited the size of datasets. Python does not suffer as much from limitations due to large data sets and is a programming language where scalability to a production level of code is facilitated. Although not as comprehensive as R, Python's machine learning library is fairly mature and comprehensive. A popular package is Scikit-learn[[5], offering many well-known algorithms such as Support Vector Machines, Random Forest, k-Means and more. With several options for implementation, and integration compatibility with ARCADE-baseline software, Python was the main programming environment utilized in this research.

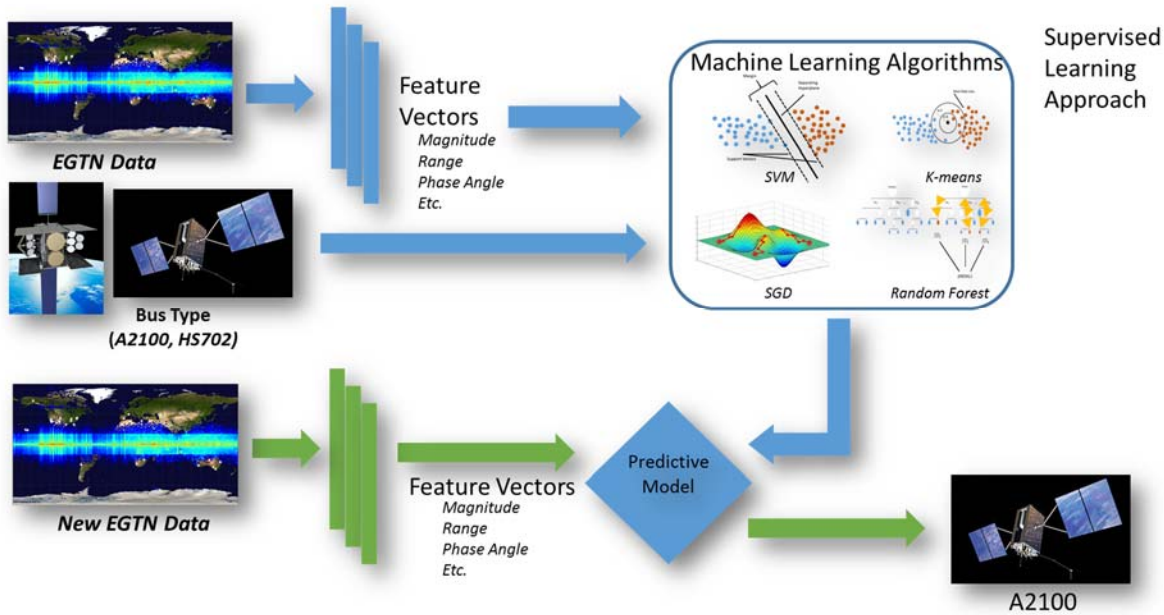

Figure 3 Machine Learning Approach

Figure 3 depicts our machine learning approach. Data is mined from the EGTN archive for pertinent observations. Depending on the type of feature being forecasted, data can be specific (i.e. for a certain time period, duration, orbit type, satellite type, etc.), or generic (i.e. all data for a month). Features such as magnitude and phase angle are then extracted from the data and input to the machine learning algorithms. A priori bus-type labels, the a priori expected outcomes, are also fed into the machine learning algorithms. In this paper, we looked at bus type (A2100 and HS702), as well as state of activity (active or inactive). The machine learning algorithms then create a predictive model. Next, the same features are extracted from newly collected data collected from the EGTN and fed into the predictive model. Finally, a resultant bus-type label is predicted, in the example shown in Fig. 3, the A2100 bus.

The principle machine learning algorithm used was Random Forest, though others were also evaluated. The data used for this analysis was taken from correlated observations of the EGTN from January 2015 to June 2016. This analysis has proven that the machine learning methods as implemented can successfully classify certain satellites, with some caveats. Between six to seven hundred observations of a particular satellite are required to have a 90% accuracy in classification on what are *a priori* believed to be active satellites. Three to four times as many observations are typically required to obtain half the accuracy on satellites believed *a priori* to be in an inactive state.

When beginning work on this research, it was unclear which data fields would be important to use in our implementation of machine learning techniques. The method of evaluation used was a tree-based feature selection method included in the Scikit-learn package. This method fits a tree classifier to the data that subsequently ranks the different fields by importance. Some of the fields ranked highly were unsurprising, while others such as magnitude uncertainty were surprisingly important. Though, the ten highest ranked features captured 50% of the information, it was decided to include all the data fields for initial testing. This was due in large part to the counter intuitive nature of some of the feature rankings.

The initial classification task was to identify the bus type of an observed satellite. Drawn at random, the initial training dataset contained more than 800,000 observations. A 15% random sampling of this training data was withheld from the model for the purpose of model validation. Sample data was collected from the most recent night's observation data and is the data that is statistically tested and classified by the machine learning algorithms. To prevent data leakage caused by what is traditionally referred to as the algorithm "memorizing" associated NORAD IDs, the field containing this information was omitted from the ML analysis, forcing the models to learn the different data features. The dataset was then scaled to ensure better model performance. Additionally, the data was filtered using NORAD IDs to ensure that test data only contained satellites that were included in the training data.

The process of constructing individual models using the Scikit-learn package is very straightforward, with all algorithms following a similar procedure. Four supervised learning algorithms were evaluated in parallel, Random Forest, Support Vector Machines, k-Nearest Neighbor, and Stochastic Gradient Descent; each using the same previously mentioned datasets. Initially, most of the algorithm parameters were fixed at default settings. With all four models trained, the validation process began by testing each model on the test data. The resultant scores were checked multiple times, and give an approximate average according to how well the individual algorithms perform. The Random Forest model was able to accurately identify 99% of the observations. Poor initial results with the SGD, SVM and k-NN models lead to an exhaustive grid search of algorithm parameters. The best results that were obtained with the SVM model used a linear kernel and resulted in 91% of the observations correctly classified. After much parameter tuning, the k-NN was also able to reach a score of 91%. The SGD model was the weakest performer only obtaining accuracy of 17% on the test data.

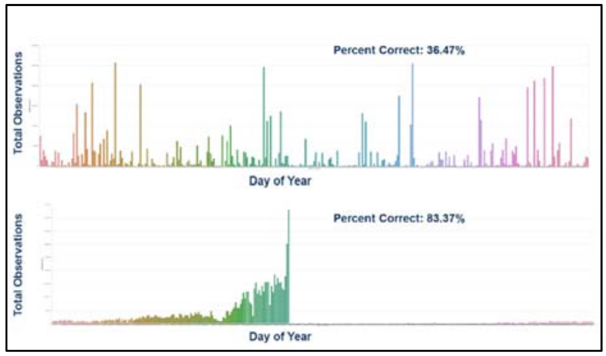

Figure 4 Comparison of different training datasets effects on test data performance.

To further verify the observed Random Forest performance, a newer test dataset was identified that was not used during training of the model and was applied. Initial tests with this second set showed very large drops in accuracy and, led to the resampling of training and testing data a number of times to confirm results. Through this repeated process, as seen in Figure 4, the importance of the time relationship between training and testing datasets was discovered. Although not unexpected[6], validation data closer in time to the training datasets returned significantly better results. Through this entire learning process the Random Forest model remained the top performing algorithm on all data. Final model performance is summarized in Figure 5, where the training dataset came from a thirty day period prior to the testing day. One notable surprise was the SGD which increased in accuracy over its validation test score. This significant drop in accuracy across all models (SGD excluded), led to the belief that the models were too complicated, causing an over fit problem.

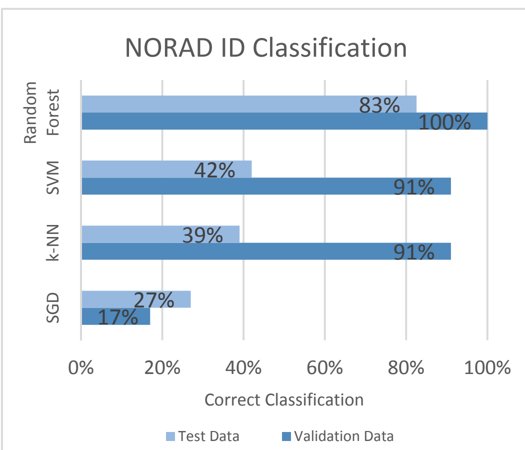

Figure 5 Initial results from four machine learning algorithms using all available fields from the EGTN data over 30 days

A separate test for other parameters was conducted to see how changing the classification goal from bus type to NORAD ID would affect model accuracy. Using the same datasets showed that the difference in accuracy was very minimal. Further research from this point on was focused on this classification task, as it allowed a more diverse number of satellites to be tested, and the datasets were quicker to compile. The Random Forest algorithm, a clear winner, became the primary algorithm for most of the follow on testing.

Three feature groups were chosen to test and evaluate their effects on the overfitting problem; i) the top thirteen most important features as ranked by the previously mentioned feature ranking algorithm, ii) six features chosen by a subject matter expert, and iii) seven features that had the lowest standard deviation between the training data and the testing data. Figure 6 shows the results of reducing the number of features. None performed as well on the test data as did the use of all available features. Surprisingly, all tests continued to perform remarkably well on validation data while not performing as well on the test data.

Uncertain how to increase accuracy any further, previous datasets were reexamined in order to potentially reveal information that had gone unnoticed. One discovery was that the satellite state (e.g. active or inactive) significantly affected the machine learning models and corresponding outcomes.

To evaluate this discovery, a small dataset of only seven satellites was taken from a list of known active satellites. These seven satellites were specifically chosen to be the most observed satellites to ensure enough observations were available. Next, fourteen satellites were chosen at random (7 active, and 7 inactive) and a Random Forest model was then constructed for each group. These two models were tested on validation data in the same manner as previously mentioned. Figure 7 shows the results from this effort. The model with the active satellites achieved a score of 99%, while the inactive satellite model scored only 62%. Attempting to determine if significantly more observations would in any way increase the dead satellite group's identification accuracy, a model was trained using only ten highly observed inactive objects. This model contained no fewer than 11,000 observations per satellite and achieved accuracy of only 47%. It is currently believed that due to inactive objects being in a more chaotic state than their active counterparts, the light-curves are more complex and lack distinct trends for model development. Additional features will likely be required to improve machine learning accuracy. This is the key takeaway - integrating existing feature extraction algorithms to provide additional training features likely provides the best approach for accuracy improvement.

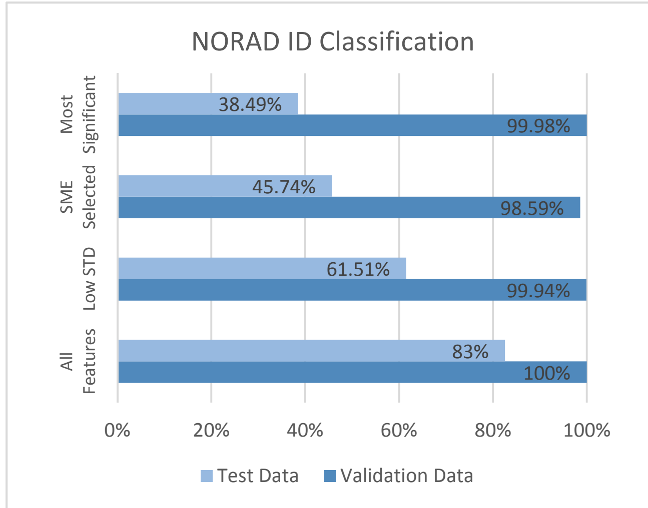

Figure 6 Results from four machine learning algorithms using a reduced set of features

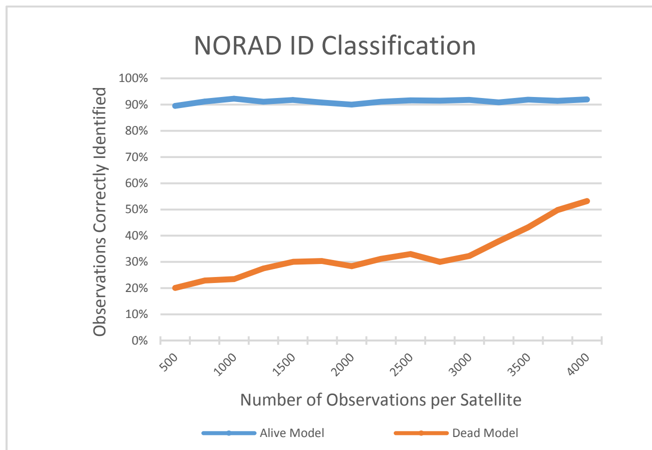

Figure 7 Results from alive and dead models

Table 2: 50th percentile results of binned visual magnitude data

```
Support Vector Machines (SVM) Accuracy: 0.89 (+/- 0.35)
Stochastic Gradient Descent (SGD) Accuracy: 0.48 (+/- 0.28)
Perceptron Accuracy: 0.62 (+/- 0.15)
K-Nearest Neighbors (KNN) Accuracy: 0.94 (+/- 0.24)
Multi Layer Perceptron (MLP) Accuracy: 0.86 (+/- 0.37)
```

Finally, classification of bus type was researched. Using data from the EGTN collected over a period of one year, 13 HS702 satellites and 20 A2100 satellites were chosen for the initial algorithm development. Visual magnitude data from these satellites were binned in one degree solar phase angle increments. Next, various percentiles of these bins were taken (25, 50, 68, 75, 97). The results shown in Table 2 are from the 50th percentile. Finally, the data was run in five different machine learning algorithms and the final model validated with a 10-fold cross validation
There are several notes on these results, the first of which is that the data sets, while using hundreds of thousands of observations from each satellites, the actual data binned solar phase angle data going into the machine learning algorithms is small, ranging from 250-300 data points per satellite or approximately one visual magnitude per solar

phase angle. Machine learning algorithms are designed to work with extremely large datasets; small datasets tend to suffer from larger variances and often cause the machine learning algorithm to over train on features that may be artificial. Second, despite the small number of data points and satellites, the accuracy of the model is high for some of the models, especially the k-NN and SVM. As noted above, the standard deviation is high due to the small amount of data being used in the model. With more satellites to train on, the accuracy will increase and the standard deviation will decrease. Third, we only explored visual magnitude as a function of solar phase angle. Other features may be more promising as a classification feature. Further, machine learning algorithms do even better with multiple features and a combination of features may prove more viable as a classifier for buses.

WEKA (Waikato Environment for Knowledge Analysis) [7] was used as another machine learning tool to explore the relationships and potential features of the ESpOC photometry data. WEKA is a free open-source tool developed by the Department of Statistics at University of Waikato in Hamilton, New Zealand. Initial classification results using WEKA are very promising. Running the same data that was used in the python code described above for Sickie-learn, WEKA created a model with a high degree of accuracy on classifying the HS702 and A2100. The model was also validated with a 10-fold cross-validation. Furthermore, the model potentially discovered a dataset that may be anomalous. Figure 8 shows the results, where data is plotted as a function of predicted class. The dots are the correct labels, with (0,0) being the A2100 and (1,1) being the HS702. The size of the X indicates the distance from the dot, or truth class. The truth points are blue and orange. One A2100 was mislabeled, as indicated by the blue X near the orange dot on the upper right. The anomalous satellite is in the bottom left, with a value -0.9, indicating that it does not fit with either class. Finally, 4 satellites were ambiguous and could not definitively be classified.

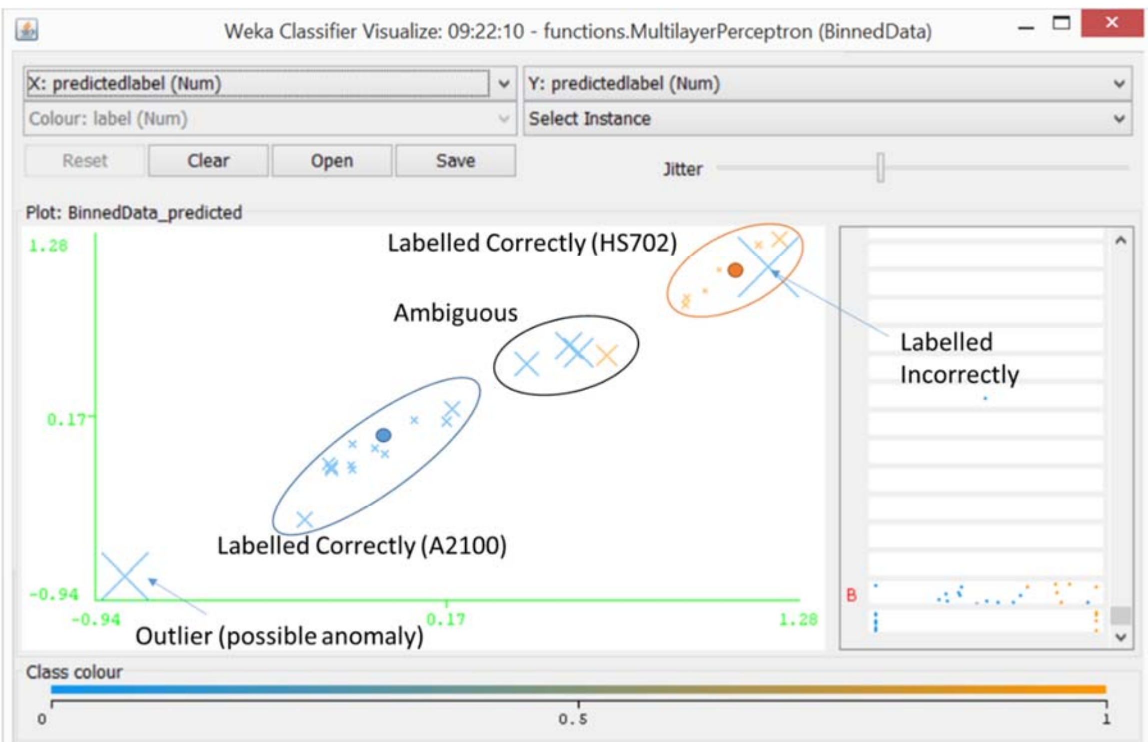

Figure 8: WEKA Classifier Visualization for Boeing 702 and Lockheed A2100

4. FUTURE WORK

The results presented in this paper are promising and show that automated machine learning on the photometric data being collected by the EGTN is possible. Future machine learning maturation work will be focused in two areas.

The first focus area is implementing these algorithms within ACDC for real-time processing. However, before this can be done, two tasks must be performed:

a. Pre-processing— As previously described, the raw data from the EGTN can be noisy and needs to be processed prior to application of the machine learning algorithms. As shown when attempting to classify the bus type, the data may need to be further processed in order to effectively represent a feature that is unique to the space object.

b. Algorithm tuning— Additional tuning of the algorithm parameters must be done in order to generate a predictive model that works well with new data. Once a model is performing sufficiently well, the algorithm can then be transitioned into ACDC and available for on-demand and persistent processing.

The second focus area is implementing feature extraction algorithms that will be able to provide derived information about the satellite that may then be added to the predictive data model as additional features. One of these feature extraction algorithms was the bus type classification. Additionally, two algorithms currently being developed for ACDC are the real-time stability (RTSE) algorithm and the Material Characterization Algorithm (MCA).

The RTSE algorithm enables notification of a user when stability of an object changes from the hypothesized mode. Currently, RTSE provides stability information to the EGTN operator (see Figure 9) via a Graphical User Interface (GUI). The stability parameter, when combined with additional features, may provide insight as to mission, activity, or anomaly resolution.

The MCA permits the analyst to determine possible material makeup of a space object, and furthermore can aid in object correlation. This algorithm utilizes multiple optical bands, and was adapted specifically to use visible (color) bands, although the original algorithm estimates multiple object features by incorporating any number of infrared bands. This algorithm can be run both persistently and on-demand, as it can run in near real-time [1]. Along with enabling the analyst in real-time, the MCA estimate can augment the machine learning algorithms previously described in this paper with additional information that may help in satellite classification.

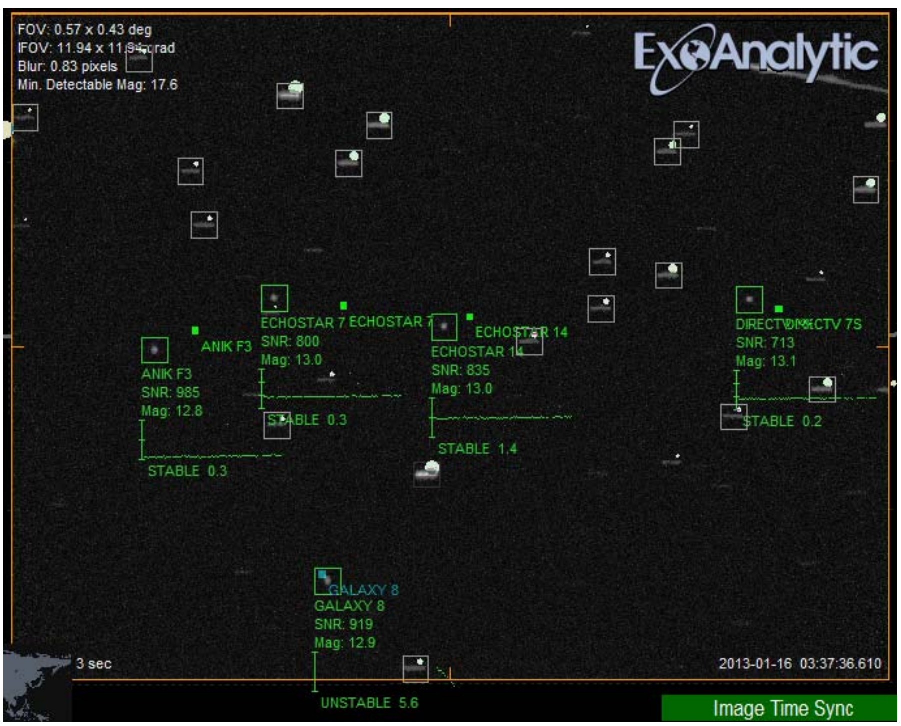

Figure 9 GEO Objects Matching Orbit-Based Hypothesis

In addition to RTSE and MCA being developed for ACDC, one additional algorithm, an empirically based "fingerprinting" algorithm, under research is a strong candidate for automated machine learning techniques. The "fingerprinting" algorithm uses individual light curves of a satellite to create a unique signature that then can be compared to other satellite light curves, similar to an expert system technique [2,6].

For GEO 3-axis stabilized spacecraft that have sun-tracking solar panels, the "fingerprint" plot, such as shown in Fig. 10 generally has a bright feature, often centered near (0,0), and gradually decreases, is dimmer, as a function of solar phase angles [6]. Notice there is typically data missing near (0,0) – this is because when the solar equatorial and declination phase angles are near zero, the spacecraft is in the shadow of the earth. Also note that to complete the "fingerprint" plot, observations need to be made at several points during the night (to cover solar equatorial phase), as well as over the course of a year (to cover solar declination phase). Even if there are several tiles of data missing from the fingerprint, it can still be useful to examine trends for both a single space object and a particular group of space objects. The important features of the fingerprints are the trends, particularly from the A2100 bus spacecraft. There is a signature peak near (0,0), as expected from the solar panels, and there is a secondary peak around (-35, 10) for all 4 spacecraft examined. This unique signature is due to solar panels that are canted relative to the sun [6]. Without observing throughout the year, it is difficult to fully realize this trend [1].

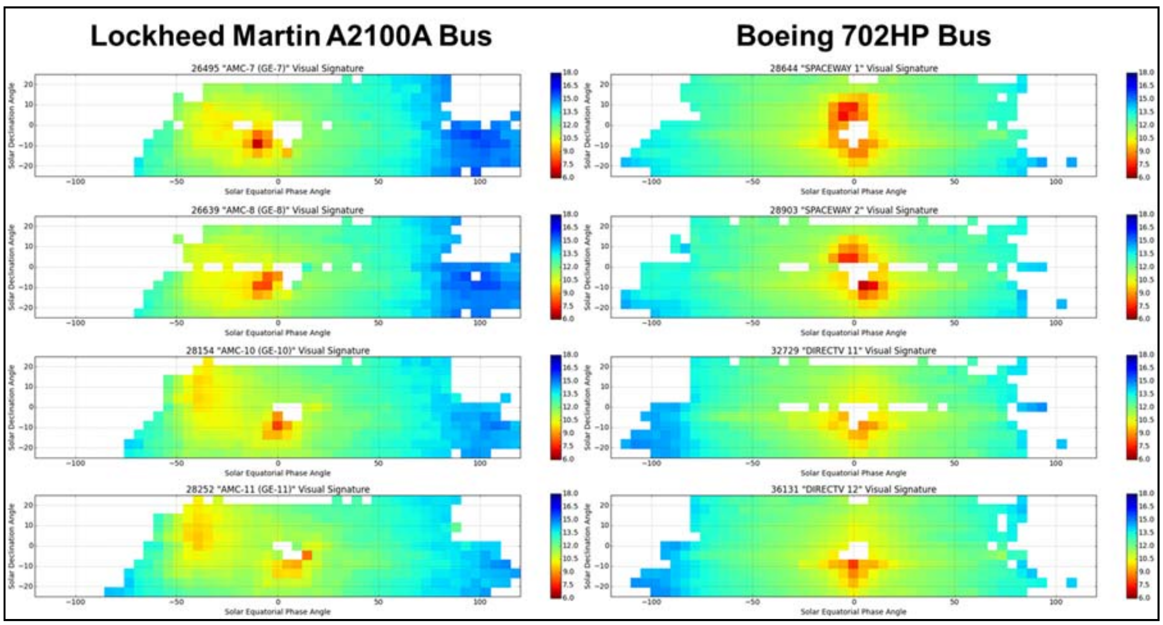

Figure 10 GEO Spacecraft Product Line Comparisons

The fingerprint plots are useful for several reasons, and fundamentally they provide the foundation for detecting state changes. With a fully-populated fingerprint plot, there is a historical basis on each space object that states can be compared to on a temporal basis. The fingerprint for each satellite could be exploited as a feature for machine learning to classify *a priori* unknown bus types in future work.

5. SUMMARY

This ongoing research demonstrated machine learning techniques on the large photometric data set collected with the ExoAnalytic Ground Telescope Network. As this research has shown additional features will likely be needed to improve machine learning accuracy and capability. Future attempts will include additional algorithms to preprocess the data before a machine learning model is trained. Moreover, future work will be focused on algorithms that provide derived features that can be used to augment the predictive data models based on the raw EGTN data. Finally, maturing the machine learning algorithms presented in this paper demonstrates a path forward for incorporating the described satellite characterization algorithms into ARCADE for further maturation relative to the JMS baseline and as part of a concept for sensor tasking using commercially available data.

6. REFERENCES

[1] Murray Krezan, J., Poole, M; *Autonomous Object Characterization with Large Datasets*, 2015 AMOS Technical Conference, September 2015.

[2] Linares, R., Jah, M. K., Crassidis, J. L., *Inactive space object shape estimation via astrometric and photometric data fusion*, Proceedings of the 22nd AAS/AIAA Space Flight Mechanics Meeting, Charleston SC, January 2012; Chaudhary A., Payne T., Lucas K., Kinateder K., Dao P., Murray-Krezan J., *Propagation of Bayesian Belief for Near-real Time Statistical Assessment of Geosynchronous Satellite Status Based on Non-Resolved Photometry Data*, Published Proceedings of AMOS Tech 2014, November 2014; Singh, N., Horwood, J. T., Aristoff, J. M., Murray-Krezan, J., *ATHENA:A Data-Driven Anomaly Detection and Space Object Classification Tool for SSA*, in Proceedings of the 26th AAS/AIAA Space Flight Mechanics Meeting, Napa CA, February 2016; Chaudhary, A.B., Payne, T.E., Gregory, S., Dao, P., *Fingerprinting of non-resolved three-axis stabilized space objects using a two-facet analytical model*, AMOS 2011 Technical Conference, Maui, HI.

[3] Murray-Krezan, J., Sabol, C., Runco, A. M., Zetocha, P., Echeverry, J., and Richard K., *The Joint Space Operations Center (JSpOC) Mission System (JMS) and the Advanced Research, Collaboration, and Application Development Environment (ARCADE)*, 2014 AMOS Technical Conference, September 2014.

[4] KDnuggets. (n.d.). Retrieved July 21, 2016, from http://www.kdnuggets.com/polls/2013/languages-analytics-data-mining-data-science.html

[5] Scikit-learn. Retrieved July 21, 2016, from http://scikit-learn.org/stable/

[6] Murray Krezan, J., Inbody, W.C., Dao, P., Dentamaro, A., Fulcoly, D., Gregory, S. A., *Algorithms for Automated Characterization of Three-Axis Stabilized GEOs using Non-Resolved Optical Observations*, 2012 AMOS Technical Conference, September 2012.

[7] WEKA. Retrieved August 10, 2016, from http://www.cs.waikato.ac.nz/~ml/weka/