Detecting Satellite Laser Ranging Station Data and Operational Anomalies with Machine Learning Isolation Forests at NASA's CDDIS

Justine Woo (1), Sandra Belvins (1), Benjamin Patrick Michael (2), Taylor Yates (1), Rebecca Limbacher (1)

(1) Science Systems and Applications INC./ NASA Goddard Space Flight Center, Code 61A, Greenbelt, MD, USA (Justine.y.woo@nasa.gov)
(2) NASA Goddard Space Flight Center, Code 61A, Greenbelt, MD, USA

Abstract

The International Laser Ranging Service (ILRS) is currently composed of 45 active satellite laser ranging (SLR) stations with more set to join the network over the next several years. Station changes and histories are logged to files, but not always in real time. Sometimes these details are not added until long after changes have been made to the station – on occasion, years later. This, in addition, to unexpected hardware errors and other system issues that are not immediately detected, impact the products generated by analysts. The ILRS Central Bureau (CB) and NASA's Crustal Dynamics Data Information System (CDDIS) have worked together to provide tools for station engineers to use. These include the creation of station plots which contain temperature and pressure information along with LAser GEOdynamic Satellite (LAGEOS) and LAser RElativity Satellite (LARES) tracking information that enable the monitoring of station performance and to determine whether the station has undergone any changes. As next steps, the CDDIS is working to enhance these station performance monitoring tools through machine learning. Isolation forest is an unsupervised machine learning algorithm commonly applied to anomaly detection. In this paper, the CDDIS details the steps taken to track anomalies within SLR station performance using isolation forest with LAGEOS and LARES satellite data.

1. Introduction

The ILRS requires stations to track their changes in station change history logs to aid analysts in identifying sources of discontinuities or changes in the station data[1]. When undocumented, this can lead to errors in data or products, and analysts, when they do catch them, will need to redo their analysis work. Therefore, early detection and notification of a change is critical to ensuring their work continues smoothly.

However, changes are sometimes logged with a delay while others are missed. Originally, some stations kept paper logs and infrequently updated the station change history log files in the CDDIS and EDC archives, limiting expediency. To simplify the process for the stations, the EDC created a webpage that allows stations to easily update and edit their station logs.

Another source of delay is the stations' ability to identify unexpected changes to hardware. Some stations do not have the resources to be able to detect issues within their hardware and software. To help stations, the ILRS website has a series of plots available for each of the stations including meteorological data, LAGEOS performance, and Satellite Data. In 2020, the Station Plots Working Group was created by the ILRS Central Bureau to update the charts and in 2021 the new charts were released on the

ILRS website[2]. However, the plots are infrequently utilized as a monitoring tool which limits their usefulness. Therefore, this paper explores the possibility of creating an alert system that reviews the stations plots for a change (anomaly) detected by an isolation forest model which then can provide a reminder for station engineers to review existing station performance monitoring plots and update the station site history logs.

2. Background

Machine learning techniques have become increasingly prevalent in the sciences due to their ability to solve a variety of problems which allow for the simplification and automation of various tasks. However, model selection remains an open-ended question for most problems as models evolve over time. Popular models for anomaly detection include k-nearest neighbor, support vector machines, decision trees, and deep learning models[3,4]. Deep learning is a particularly popular topic currently and although it has plenty of applications in anomaly detection and geodesy, it tends to require large data sets and incurs high runtimes. In addition, current research shows that tree-based models continue to perform 20-30% better on medium-sized (~10,000 samples) tabular data[5].

To detect anomalies in station data, the CDDIS has chosen to use isolation forest models which have been successfully applied to predictive maintenance, fault prevention, and automation. Isolation forest models are derived from decision trees which make decisions based on inductive reasoning. Decision trees classify instances by sorting them through a series of if-else statements based on attributes of an instance that travel down from the root of a tree to leaves. The logic of decision trees is easy to follow because their decisions can be printed as if-then rules for researchers to evaluate.

To ensure robustness, ensembles of trees are built for a given dataset, called forests. Predictions are made by averaging the predictions of each tree. Isolation forest models were first introduced in 2008 and, using the logic of decision trees, anomalies are detected based on how far from the root a particular data point is set; when a data point is close to the root, it is detected as an anomaly.[6] The model is an unsupervised learning technique, meaning the dataset used to train the model does not have explicit labels for whether the data is normal or anomalous. This prevents the model from being accidentally trained to recognize only a subset of issues that may be more common. In addition, when working with incomplete station change history logs or ones with possible missing records of changes, the model will not be accidentally trained to skip anomalies that are similar to ones that were not logged.

3. Dataset and Methods

Isolation forest models were built using data retrieved from the daily files available at the CDDIS[7] from Yarragadee (YARL) station used as a starting point to determine the feasibility of this type of program. YARL's data doesn't have much scatter and the station is the highest performer (obtains the most passes). The models built take the past 90-days of data for LAGEOS and LARES as a training set. LAGEOS and LARES were chosen due to their consistent orbits. From there, test data from the next seven days are run against the model to check if an anomaly is detected. 49,962 data points were evaluated from 2013/05/01 to 2022/07/19 for 3,579 unique days. On average, 14 data points exist for each day although there is variation over time and seasons. Some data points were not included such as those with incomplete information.

The station plots on the ILRS website for LAGEOS and LARES data include the average bin RMS, calibration RMS, system delay, observations per normal point, and full-rate observations per pass[8]. Initial tests were run with all these features but provided an excessive number of false positives. To target changes that could be linked to specific changes within the station change history logs, the features were narrowed down with the input of station engineers. The following features were ultimately selected for anomaly detection for laser and receiver subsystems: 1) calibration RMS of raw system delay (ps), 2) median of the calibration RMS, 3) median of the average bin RMS from the mean raw accepted time-of-flight minus the trend function (ps) over the course of a day 4) system delay peak (mean value (ps)) of the calibration and 5) satellite identifier. Medians were added to the data set to provide a stronger pattern for the model to detect and to reduced scatter or noise. The satellite identifier was added because the model should recognize it as an unimportant feature; it can, therefore, act as a check on the whether the model reflects current knowledge of feature importance.

With the model and features selected, model parameters were adjusted to improve performance. Python packages for machine learning simplify the process of using machine learning because models do not have to be built from scratch. For isolation forests, the CDDIS utilized scikit-learn[9]. The following hyperparameters were tested while default values were used for other options available:

- n_estimators = number of base estimators in the ensemble; i.e. the number of decision trees generated to create the forest
- max_samples = number of samples to draw from the training data to train each base estimator
- contamination = amount of contamination of the data set, i.e. the expected proportion of outliers

Additional parameters were created to reduce the number of false positives. Rather than using the full dataset available and generating a single model, models are created to look a set number of days, so predictions are made based on more recent and relevant data. This ensures that minor changes are detected. In addition, rather than sending an email for each anomaly detected, the model is set to check if an anomaly is detected over multiple days before final determination.

- daysBack = the number of days backward the model should review before making a prediction; i.e. the training data set
- testDays = the number of days that the model should run predictions on, i.e. the test data set
- detectDays = the minimum number of test days that have an anomaly where an alert will be triggered

An additional factor that determined whether a test should be run was the amount of sufficient data. For example, tests were not run if the model was created using less than 100 data points or the number of data points to test was under 25.

All the parameters were varied to test how well they aligned with the station change history log. In the final the model framework, isolation forest predictions were based on the past 90-days' worth of data (daysBack) and tested against the following 7-days (testDays). The number of estimators (n_estimators) was set based on daysBack multiplied by 2. Therefore, a total of 180 trees were generated per forest. For the maximum number of samples used to generate each tree, the number of data points used is the average number of passes per day, multiplied by the number of days reviewed by the model*20%. 20% was chosen because it is a standard value used to generate models for predictions in machines learning. When less data are available it may be advantageous to use a larger percentage of the data for model development.

Contamination was set extremely low to prevent the triggering of false-positives. However, in doing this, more discrete anomalies and changes listed in the station change history logs were not detected. To correct this, the algorithm checks if anomalies are detected in three days (detectDays) out of seven (testDays) and does not take into account the actual probability returned.

For each test set, if three days or more were labeled as having anomalous data, the calculated RMS, calibration RMS, and system delay plots were created for review. An example is shown in Figures 1-3 with data from the anomalous dates highlighted.

Figures 1-3: Plot of Calculated RMS, Calibration RMS, and System Delay training and test data where an anomaly is detected (circled)

This example shows a detected anomaly that was not listed in the station change history log but shows a visible jump in the system delay. Although an anomaly is detected, not all system delay values within that change were listed as anomalies because the contamination value is set extremely low and the other features also have an impact on whether each individual point is detected as anomalous.

It is possible with individual trees to show the decision-making process; however, for forests, Shapley (SHAP) values are commonly used for machine learning explainability. SHAP values are the average expected marginal contribution of each feature after all possible combinations have been considered. It is a widely used approach from cooperative game theory for machine learning explainability[10]. In Figure 4, the primary indicator that a station has changed (anomaly) is the system delay. As a sanity check, the satellite identifier is included because this should not impact the data.

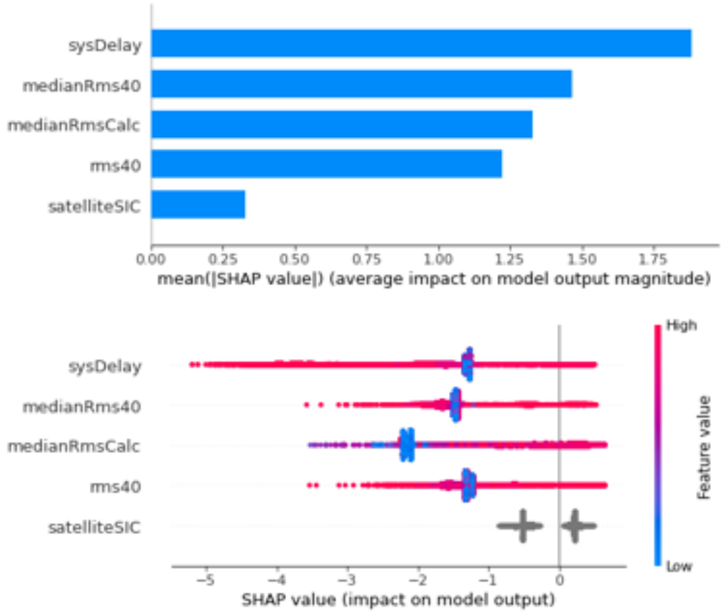

Figure 4: SHAP values for sample prediction

4. Results

A total of 271 distinct days were detected as having anomalous data. A single day can trigger multiple anomalies given that models are built in succession, with the following model built after shifting a day. When the model is run on a specific day, a probability that the date contains an anomaly is given as a percentage. Figure 5 shows the sum of those probabilities for each day tested. Where anomaly count is equal to 7, for all the models that the day was tested on, the date was detected to have an anomaly with 100% certainty, which reflects the highest probability.

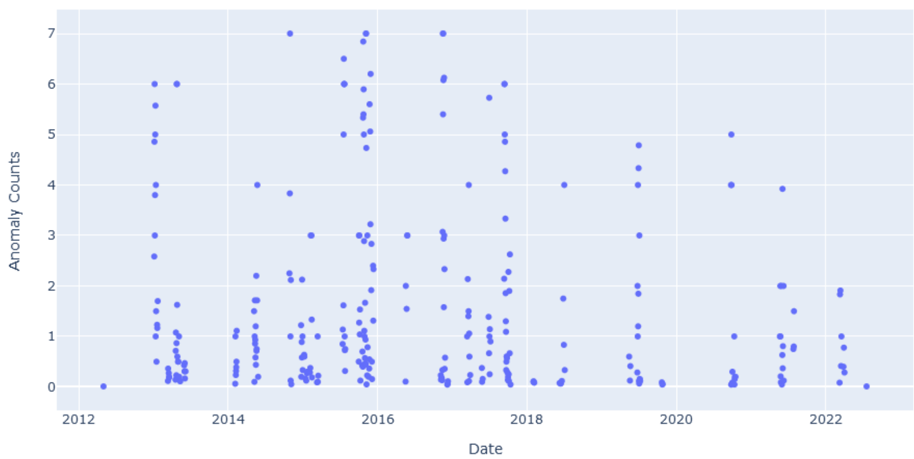

Figure 2: Total anomalies detected with the model prediction plotted against the session datetime

To verify the results, each prediction was compared against the YARL station change history log. All records pertaining to changes with the laser and receiver systems where the impact factor is more than negligible were detected. 99% of the correctly detected changes were detected the same day as was recorded in the station log.

Figure 5 also shows how a single change can affect data for a week or two, resulting in anomalies being detected for a while afterward. This dragging effect is overcome by only triggering an alert when one has not been triggered the previous day. Therefore, although 45% of the detected anomalies do not have a record in the station log, they can be traced back to an earlier change and only 17% would generate an email, that is 21 over 10 years. It is also possible that station changes were not recorded in the station log, however, of the cases reviewed, this only occurred once for YARL.

4. Discussion

The results from YARL are promising but additional tests with other stations need to be incorporated. It needs to be determined if several model frameworks can exist based on stations with similar features or if each station is unique enough to warrant its own model framework. The probability that a single model framework will work for all stations is quite low given that the model framework will need to be adjusted based on session availability, contamination levels within the data, noise within station data, and station hardware and software differences.

Session availability varies by station for LAGEOS and LARES data. The ILRS requirement states that stations only need to track 4 passes per satellite per week for the geodetic satellites[11]. YARL exceeds this, tracking at least one of the satellites each day if not all of them. For the days tested only 50 were skipped of 3,629 days because there was not enough data. However, some stations meet the requirements by tracking all the satellites a day or two out of the week. This reduces the chance for early and accurate detection. There will be a need to lower the number of sessions required for training and testing leading to less accurate models. It is possible to extend the period reviewed to get more training data or to add additional satellites to increase the number of sessions but both these will result in less accuracy and delays in anomaly detection. In addition, parameters such as daysDetected will need to be removed resulting in increased false positives.

Isolation forests also have a limitation in model transferability given that they can only be transferred if the stations have similar contamination levels. This can vary based on the amount of noisy data or changes each station makes.

The current model is also limited to the amount of features tested because it was built so that comparisons to the existing station plots could be made – i.e. to easily determine if the detections are correct. Adding more features such as RMS50, skew kurtosis, peak-mean, and pressure can improve the accuracy of the model and perhaps detect changes in other subsystems.

These changes must be investigated before an automated program, which generates models for each station, can be released. For automation, the model weights, durations, and features can be updated based on the percentage of correct detections, setting a maximum for the percentage of false-positive emails that are sent, and comparisons against the site history log where applicable.

Acknowledgements

I'd like to thank the Station Plots Working Group members for their input on the features important to anomaly detection from when the station plots where being reformulated. I'd like to give a special thanks to Van Husson for his input in reviewing the initial software outputs and his station expertise.

I'd also like to thank the SSAI Deep Learning Academy, especially Brandon Smith for his help in reviewing the machine learning software and for his input on the clarity provided from its inputs and outputs.

References

1. Procedures for Tracking ILRS Station Changes. https://ilrs.gsfc.nasa.gov/network/site_procedures/configuration_files.html (2019) Accessed 27 December, 2022
2. Overview of Active Station Plots https://ilrs.gsfc.nasa.gov/network/stations/active/overview_of_station_plots.html (2021) Accessed 27 December, 2022
3. Omar, S., Ngadi, A., Jebur, H. Machine Learning Techniques for Anomaly Detection: An Overview. In: International Journal of Computer Applications. Volume 79. 2013
4. Comparing anomaly detection algorithms for outlier detection on toy datasets. https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_anomaly_comparison.html#sphx-glr-auto-examples-miscellaneous-plot-anomaly-comparison-py Accessed 19 July, 2022
5. Grinsztajn, L., Oyallon, E., Varoquaux, G. : Why do tree-based models still outperform deep learning on typical tabular data? (2022) Accessed 23 November, 2022 https://doi.org/10.48550/arXiv.2207.08815
6. F. T. Liu, K. M. Ting and Z. -H. Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.
7. International Laser Ranging Service (ILRS), SLR daily normal point data, Greenbelt, MD, USA:NASA Crustal Dynamics Data Information System (CDDIS), Accessed 19 July, 2022 at doi:10.5067/SLR/slr_data_daily_npt_001.
8. Yarragadee: LAGEOS performance https://ilrs.gsfc.nasa.gov/network/stations/active/YARL_station_info.html?LAG (2021) Accessed 19 July, 2022.
9. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
10. An introduction to explainable AI with Shapley values. https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html (2018) Accessed 19 July, 2022.
11. System Performance Standards (2015) https://ilrs.gsfc.nasa.gov/network/system_performance/index.html (2020) Accessed 28 December, 2022