Operating INTEGRAL through dynamic radiation environment with machine learning

Tomaž Stepišnik[1,2,*], **Timothy Finn**[3], **Nikola Simidjievski**[1,2,4], **Richard Southworth**[3], **Guillaume Belanger**[5], **José Antonio Martínez Heras**[6], **Matej Petković**[1,2], **Pančе Panov**[1,2], **Sašo Džeroski**[2], **Alessandro Donati**[3] and **Dragi Kocev**[1,3*]

[1]Bias Variance Labs, Ljubljana, Slovenia
[2]Jožef Stefan Institute, Ljubljana, Slovenia
[3]ESOC, European Space Agency, Darmstadt, Germany
[4]University of Cambridge, United Kingdom
[5]ESAC, European Space Agency, Madrid, Spain
[6]Solenix Engineering, Darmstadt, Germany

Abstract

The INTEGRAL spacecraft orbits Earth since 2002 in a highly elliptical orbit, passing through the Van Allen belts – areas with high-energy ionised particles that can damage the spacecraft's on-board equipment. An essential part of mission planning and operation of INTEGRAL is thus the prediction of its radiation belts entry and exit times. We propose a novel compact representation of the data and evaluate its potential using several machine learning methods. The experimental validation identifies gradient boosted trees with quantile loss as the best performing method. By using our approach, INTEGRAL can perform 2 additional hours (on average) of scientific measurements per orbit (with adjustment for uncertainty at the 95th percentile). This approach protects INTEGRAL from damages and improves its scientific return at the same time.

1 Introduction

INTEGRAL (INTErnational Gamma-Ray Astrophysics Laboratory) is an astronomical observatory of the European Space Agency (ESA) that orbits Earth since 2002. It is in a highly elliptical 64-hour period orbit, spending most of its time (approximately 55 hours per orbit) observing high energy sources, such as black holes, neutron stars, active galactic nuclei (AGN), regions of nuclear emission lines, and other exotic astronomical bodies. It also detects γ-ray bursts (GRBs) in the dozens per year, and in real time. The highly eccentric nature of INTEGRAL's orbit, with an apogee height of approximately 140000 km and a perigee of approximately 6000 km, allows it to spend most of its time out of the Earth's Van Allen radiation belts [Li and Hudson, 2019] (Figure 1) – located in the innermost region of the Earth's magnetosphere. However, in the remaining time, the crossing through these belts [Walker and Palmer, 2012] poses a great threat to the endurance of INTEGRAL and its scientific payload.

The long-term success of the INTEGRAL mission depends on accurate estimations of its radiation belts entry/exit times, which allow for precise planning of orderly deactivation and activation of the on-board equipment. The on-board autonomy function implemented in INTEGRAL's central data management unit issues environmental information to each instrument every 8 seconds. Based on this information, the on-board instruments respond to the predicted entry/exit times (typically computed several months in advance) and ensure an orderly shutdown of the instruments before the spacecraft's entry in the belts, and a timely restart after the spacecraft's exit from the belts. Without accurate predictions, the instruments are often forced to perform emergency shutdowns and restarts based on the IREM readouts alone. This is followed by a lengthy recovery protocol and delayed operation of the instruments.

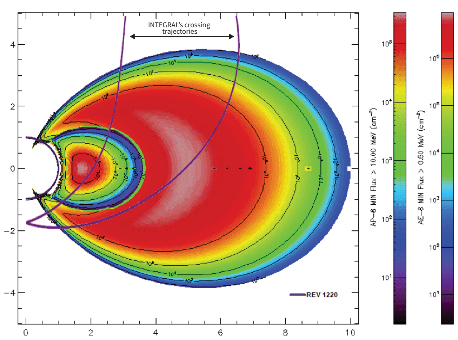

Figure 1: Top-view projection of INTEGRAL's trajectory through the belts as presented in Walker and Palmer [2012].

In this work, we address the task of performing an accurate and efficient end-to-end prediction of INTEGRAL's entry/exit times of crossing Van Allen Belts. We also provide safety margins on the obtained predictions by utilizing quantile regression. The results from the experimental evaluation show that using our pipeline yields more than *200 additional hours* (over the course of 100 revolutions) for doing science.

Most closely related to our work are those of Métrailler *et al.* [2019] and Finn *et al.* [2018], which address the task of modeling INTEGRAL's crossing through the belts. The former uses data-driven modelling to construct a dynamical 3D volume model for the Van Allen belts, which can be used to predict the entry and exit times. It uses the radiation flux

*Contact Authors: tomaz@bvlabs.ai, dragi@bvlabs.ai

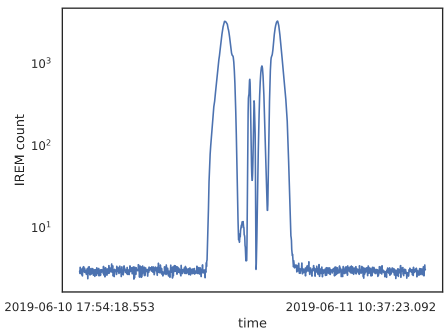

Figure 2: IREM counts for one revolution of the INTEGRAL spacecraft. The spike in radiation level indicates that the spacecraft is inside the Van Allen belts.

measurements data collected from the radiation monitors onboard INTEGRAL and XMM-Newton. The resulting model captures the long-term trends and seasonal variations very well, however, it struggles with the peak amplitudes. The latter, a predecessor to our work, considers machine learning for predicting INTEGRAL's entry/exit times, making use of random forests for predicting IREM counts and in-belts indicators from positional data only. We improve upon this work in several ways by introducing other data representations as well as evaluating more suitable methods that ultimately lead to better predictive performance. The proposed approach also significantly improves upon the performance of the model currently employed that is based on fitting a sinusoidal curve and applying an amplitude-dependent safety margin modulated around two standard deviations of the fit residuals. Currently, the pipeline proposed in this work is in 'preparation and testing for deployment' within ESA's INTEGRAL Mission Planning & Spacecraft Operations.

2 Machine learning pipeline

To determine when the spacecraft enters the Van Allen belts, we rely on the onboard radiation monitoring instrument (IREM) measurements every 8 seconds. Figure 2 shows the IREM counts for one revolution of the spacecraft. Because they can be very noisy, we take median values from bins with a coarser time granularity (5-15 minutes). Determination of entry into and exit from the belts is based on a threshold IREM count of 60.

The orbit of each revolution of the spacecraft is defined by 12 orbital elements Finn *et al.* [2018]. Additionally, we take into account the eclipse times when the spacecraft is shadowed from the Sun by the Earth or the Moon. The orbital elements and eclipse times are available for several months into the future, and are the basis from which we engineer features that the models use to predict entry/exit times. We transform all timestamps to *phase* values relative to the current revolution: $\mathtt{phase}(t) = \frac{t - \mathtt{perigee\_time}}{\mathtt{period}}$, where perigee time and period are revolution-dependent. The phase value goes from 0 at perigee to 1 at the next perigee.

We consider two data representations – *positional* and *per-revolution*. In the *positional representation* (details in [Finn *et al.*, 2018]), the data is organized in a time series where examples describe the state of the spacecraft every couple of minutes (5 - 15 min). The state of the spacecraft is captured with 23 features: current timestamp (1), orbital elements (12), position and velocity in geocentric-equatorial coordinate system (6), altitude (1), and binary indicators denoting eclipses (3). With this representation, we can predict the IREM counts at different timestamps. Alternatively, we can also threshold the IREM counts and get binary indicators that tell us whether the spacecraft is inside the Van Allen belts at the corresponding timestamps. We can use both predicted IREM counts and in-belts indicators to determine entry (and exit) times by finding the timestamp at which the counts rise above (or drop below) the threshold or when the in-belts indicators switch values. In the *per-revolution* representation, the data is organized in a time series of revolutions, where each example describes one revolution of the spacecraft using the 12 orbital elements, together with the times when the spacecraft enters and exits eclipses. This yields a total of 18 features. Here we can directly predict the entry/exit altitudes (or times) for a given revolution. This gives us two target variables and we can treat the problem as a multi-target regression task. Each revolution takes approximately 64 hours. In the positional representation with a 15-minute granularity, there are 264 timestamps (examples) during each revolution, which is a single example in the per-revolution representation. The per-revolution representation is therefore much more compact.

We consider several machine learning methods: k nearest neighbor regressor (KNN), random forest ensembles of regression trees (RF), extreme gradient boosting ensembles of regression trees (XGB), gradient boosting ensembles of regression trees with quantile loss (GB), fully connected neural networks (FCNN), and recurrent neural networks (RNN) with gated recurrent units.

For KNN, RF, and GB methods, we use the implementations as provided in the *scikit-learn* Python library Pedregosa *et al.* [2011]. For XGB, we use the *xgboost* Python library Chen and Guestrin [2016]. We implemented the neural network models in the Pytorch framework Paszke *et al.* [2019]. While RNNs, in particular, are well suited for time series data by design, the remaining methods require additional engineering for taking the temporal aspect into account. To this end, we add additional historical information to each example, i.e., each example has access to the features of the previous n examples and the targets of the previous m examples. We call the value n *feature history* and value m *autoregression history*. For all methods, features and targets are standardized (using only information from the learning data) prior to model learning. In the end, the model predictions are inversely transformed to get values on the original scale.

3 Experimental evaluation

We perform the evaluation in two stages: 1) we select the optimal parameters of the machine learning methods; 2) we compare the different methods with the selected optimal parameters to each other and the model currently used by the mission planning team. For the first stage, we use the data from 2015-03-19 to 2020-04-09 (695 revolutions). Within this period, we select 100 cut-off points at random. For each cut-off point, a method learns on data up to that point and makes predictions from that point onward. The methods make predictions for 35 revolutions after the cut-off point (cca 3 months). These predictions are used to evaluate different parameter configurations of the methods. We optimize both data preprocessing parameters (representation, granularity, feature/autoregression history) and method parameters.

For the second stage, we use the data from 2020-04-09 to 2020-12-31 (100 revolutions). We want to test how well the optimal parameters selected in the first stage generalize to this validation period. In this stage, we simulate how the models will be used in mission planning. Each method is asked to produce a model every month, which is then used to make predictions for the upcoming month. Because entry and exit altitudes are what the mission planning team is ultimately interested in, the models are evaluated by calculating the root mean squared error (RMSE) of the entry and exit altitudes determined from their predictions.

We obtained the best results with the GB method. With optimized parameters, its entry altitude RMSE was 5998 and exit altitude RMSE was 2941. In comparison, the entry and exit altitude RMSEs of the currently used sinusiod model was 6028 and 8676, respectively. The parameter optimisation also revealed that per-revolution representation is not only more efficient but also produces more accurate models (Figure 3).

Compared to the currently used model, our approach is much more accurate in predicting the exit altitudes, whereas the errors of entry altitude predictions are somewhat similar. A plausible reason for this is that, in the specific validation period considered, the entries are especially volatile with above average frequency of outliers and missing values (Figure 4). On the other hand, exit altitudes are relatively stable in this period, but slightly lower than in the past. Our models are capable of capturing these changes, in contrast to the sinusoidal model which greatly overestimates the exit altitudes.

Another advantage of GB with quantile loss is that we can take into account that some errors are more costly than others. For instance, if the predicted altitude is too high (for entry or exit), the instruments on the spacecraft will simply shut down too early (or restart too late) therefore losing some science time. However, if the predicted altitude is too low, the instruments will shutdown earlier than anticipated (due

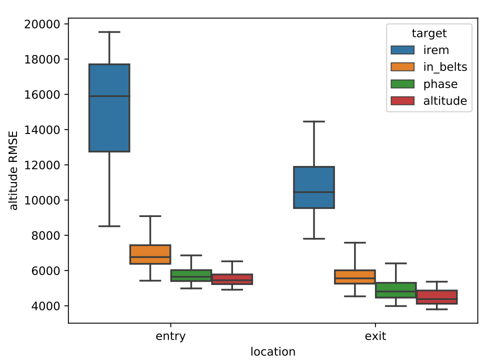

Figure 3: Performances of different parameter configurations of RF models grouped by the predicted target. The bottom whiskers show the minimum RMSE, the top show the 95th percentile.

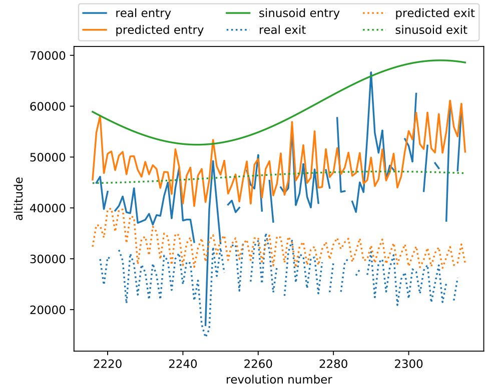

Figure 4: Comparison of real entry and exit altitudes to the upper bounds obtained with the currently used sinusoid model and the GB method. Lines showing real altitudes are discontinued where targets are missing due to IREM crashes.

to emergency shutdowns) which interrupts ongoing measurements that are carefully planned months ahead. Hence, it is important to minimize the risk of such occurrences.

Currently, the mission planning team adds a fixed margin to the predictions of the sinusoidal model. For the GB method, we can modify the percentile that the model learns to predict – instead of the 50th percentile (the median), we can predict the 95th percentile. This gives predictions close to the upper bounds of the possible altitudes and minimizes the risk of emergency shutdowns. Figure 4 show the real altitudes together with the upper bounds obtained with the sinusoidal and the GB models for entries and exits, respectively.

Performing mission planning based on the predictions of the currently used sinusoid model would lead to loss of 294.5 hours of science time over the validation period. Because the approach is quite conservative, the ongoing experiments would be interrupted only once. On the other hand, relying on the predictions of the GB method would lead to loss of only 94.3 hours of science time. This means that over the course the validation period of 100 revolutions, our approach would recover over 200 hours of science time equivalent to more than 3 full INTEGRAL revolutions. Since the margin of our approach is less conservative, the predicted altitude would be too low on 13 occasions (for 12 entries and 1 exit).

4 Conclusions

This paper describes our approach to predicting INTEGRAL's entry and exit times from the Van Allen belts. The results of the empirical analysis comparing several machine learning methods showed that the best performing method on these tasks are the gradient boosted trees with quantile loss. In particular, compared to the model that is currently used to predict INTEGRAL's entry and exit times, our model (with adjustment for uncertainty at the 95th percentile) provides on average 2 additional hours per revolution for performing scientific measurements, i.e., a gain of 3 full INTEGRAL revolutions over the course of 100 revolutions. These improvements facilitate better mission planning and optimal use of the on-board scientific equipment, thus further increasing the scientific return of the spacecraft.

References

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

Timothy Finn, Redouane Boumghar, Jose Martínez-Heras, and Antigoni Georgiadou. Machine learning modeling methods for radiation belts profile predictions. In *Proceedings of the 2018 SpaceOps Conference*, pages 1–11. American Institute of Aeronautics and Astronautics, Inc., 2018.

W. Li and M.K. Hudson. Earth's van allen radiation belts: From discovery to the van allen probes era. *Journal of Geophysical Research: Space Physics*, 124(11):8319–8351, 2019.

Lionel Métrailler, Guillaume Bélanger, Peter Kretschmar, Erik Kuulkers, Ricardo Pérez Martínez, Jan-Uwe Ness, Pedro Rodriguez, Mauro Casale, Jorge Fauste, Timothy Finn, Celia Sanchez, Thomas Godard, and Richard Southworth. Data-driven modelling of the van allen belts: The 5drbm model for trapped electrons. *Advances in Space Research*, 64(9):1701–1711, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, and Trevor et al. Killeen. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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

M. J. H. Walker and J. B. Palmer. *Integral: Investigation into Van Allen Belt and Geotail Crossings*, pages 573–586. 2012.