GalaxAI: Machine learning toolbox for interpretable analysis of spacecraft telemetry data

Ana Kostovska[1,2] , **Matej Petković**[1,2] , **Tomaž Stepišnik**[1,2] , **Luke Lucas**[3] , **Timothy Finn**[4] , **José Martínez-Heras**[5] , **Panče Panov**[1,2] , **Sašo Džeroski**[2] , **Alessandro Donati**[4] , **Nikola Simidjievski**[1,2,6] , **Dragi Kocev**[1,2]

[1]Bias Variance Labs, Ljubljana, Slovenia
[2]Jožef Stefan Institute, Ljubljana, Slovenia
[3]LSE Space GmbH, Gilching, Germany
[4]ESOC, European Space Agency, Darmstadt, Germany
[5] Solenix Engineering, Darmstadt, Germany
[6]University of Cambridge, Cambridge, UK

Abstract

We present GalaxAI - a versatile machine learning toolbox for efficient and interpretable end-to-end analysis of spacecraft telemetry data. GalaxAI employs various machine learning algorithms for multivariate time series analyses, classification, regression and structured output prediction, capable of handling high-throughput heterogeneous data. These methods allow for the construction of robust and accurate predictive models, that are in turn applied to different tasks of spacecraft monitoring and operations planning. More importantly, besides the accurate building of models, GalaxAI implements a visualisation layer, providing mission specialists and operators with a full, detailed and interpretable view of the data analysis process. We show the utility and versatility of GalaxAI on two use-cases concerning two different spacecraft: i) analysis and planning of Mars Express thermal power consumption and ii) predicting of INTEGRAL's crossings through Van Allen belts.

1 Introduction

Spacecrafts operate in extremely challenging and unforgiving environments. This calls for careful planning of their operations and close monitoring of their status and health [15]. The spacecraft monitoring includes analysing housekeeping telemetry data that measure and describe the spacecrafts status, its activities and its environment. These include temperature values at different locations, radiation values, power consumption estimates, status/command execution of active on-board equipment, performed computational activities [22; 4; 1; 21; 14; 10; 20]. Analysing telemetry data is complex and nontrivial, since such data is typically high dimensional, multimodal, heterogeneous, with temporal dependence, has missing values, and contains obvious outliers[22]. Based on the analysis of telemetry data, the spacecraft mission-planing and operations teams make decisions about spacecraft next operations - *what* activities it will perform (in terms of its mission) and *when* it will perform them.

In this paper, we present GalaxAI - a machine learning toolbox for efficient and interpretable end-to-end data analysis of spacecraft telemetry data. We showcase its potential by analysing telemetry data of two spacecraft operated by the European Space Agency: Mars Express and INTEGRAL.

Mars Express (MEX), a long-lasting mission of the European Space Agency, has been exploring Mars since 2004. It is responsible for a wealth of scientific discoveries, including evidence of the presence of water (above and below the surface of the planet), an ample amount of three-dimensional renders of the surface, and a complete map of the chemical composition of Mars's atmosphere. The scientific payload of MEX consists of 7 instruments, which need to be kept within their operating temperature ranges (from room temperature for some instruments to -180^oC for others). To maintain these temperatures, the spacecraft is equipped with an autonomous thermal system composed of 33 heater lines that consume a significant amount of the total generated electric power - leaving a fraction to be used for science operations. Hence, given the age and the current condition of MEX, monitoring and optimally planning this consumption has a direct consequence on the longevity of the spacecraft and its mission [13; 3; 17; 2; 19].

INTEGRAL is a space observatory designed to monitor and detect gamma-rays with high-sensitivity. Since its launch in 2002, it has been responsible for detecting iron quasars, investigating high energy gamma-ray burst as evidence of black-holes, supernovae remnants and active galactic nuclei (AGNs), as well as providing imaging and spectroscopic observations of astronomical events in both the X-ray range and optical wavelengths. During its 64-hour orbit around Earth (with apogee of $\sim$140 000 km and perigee of $\sim$6 000 km), INTEGRAL passes through the Van Allen radiation belts, where radiation levels are high enough to potentially damage the on-board equipment. While the spacecraft is equipped with radiation sensors, these operate autonomously, and are used for emergency instrument shutdowns, which are followed by lengthy recovery procedures. Accurately modeling and predicting the spacecraft's position w.r.t these radiation belts is important. This allows for better control over activation/deactivation of on-board instruments and ultimately leading to optimal scientific output [7].

2 Description of GalaxAI

GalaxAI follows a two-layer design, consisting of a back-end and a front-end layer (Figure 1). The cornerstone of GalaxaAI, its machine learning (ML) framework, is implemented as a part of the back-end layer. Besides modularity and easy-maintenance, such implementation also allows certain data/compute intensive ML routines to be automated and executed on dedicated computing infrastructures.

The ML framework consists of three major parts: (1) data preprocessing, feature engineering and selection, (2) model construction (learning), and (3) making predictions with a learned model. The first part includes various data preprocessing techniques and feature engineering algorithms designed and employed to pre-process the raw telemetry data pertaining to a particular spacecraft. The second part focuses on learning predictive models suitable for a considered data analysis task. The third part of GalaxAI focuses on making predictions and visualising the findings. These range from simply plotting the predicted values to more sophisticated analysis of the utility and relevance of the features used in the model construction phase.

GalaxAI is also fully operable from a front-end layer through a Graphical User Interface (GUI). This enables users without any particular expertise in ML to execute different parts-of or the complete data analysis pipeline. The interface employs React [6] front-end library and Electron [8], a platform for building desktop applications. For interactive visualizations, GalaxAI employs the Plotly.js library [11].

2.1 GalaxAI-MEX

Input data. GalaxAI-MEX processes six heterogenous types of data. These include *solar aspect angles (SAA)*, *detailed mission operaton plans (DMOP)*, *flight dynamics timeline (FTL)*, *various events (EVT)*, *long-term data (LT)* and *power data (PW)*. Details describing types of data are given by [17].
Data preprocessing. Given the heterogeneous raw data, the preprocessing within GalaxAI-MEX includes data alignment, feature construction, aggregation of the power data, and data cleansing. The data is first aligned to a given time-granularity, as the entries from various data files are recorded at irregular paces. The next step of *feature construction* creates features used for learning the predictive models. The *aggregation of the power-data* includes computation of the average electrical current (e.g., for every 15 minutes). Finally, *data cleansing* removes/imputes records with missing values from the data.
Learning models. For learning predictive models, GalaxAI-MEX implements several ML methods. Namely, it implements unified wrappers for XGBoost [5], PCT-based ensembles [12], (deep) fully-connected neural networks [9], as well as for all models implemented in the scikit-learn toolbox [16]. Moreover, it includes feature ranking methods to provide better understanding of the models and the predictions. In particular, it provides three feature ranking scores [18] that calculate the feature importance: (1) random forest mechanism, (2) GENIE3, and (3) Symbolic scores.
Making predictions. At the final stage, the constructed predictive models are employed for making predictions.

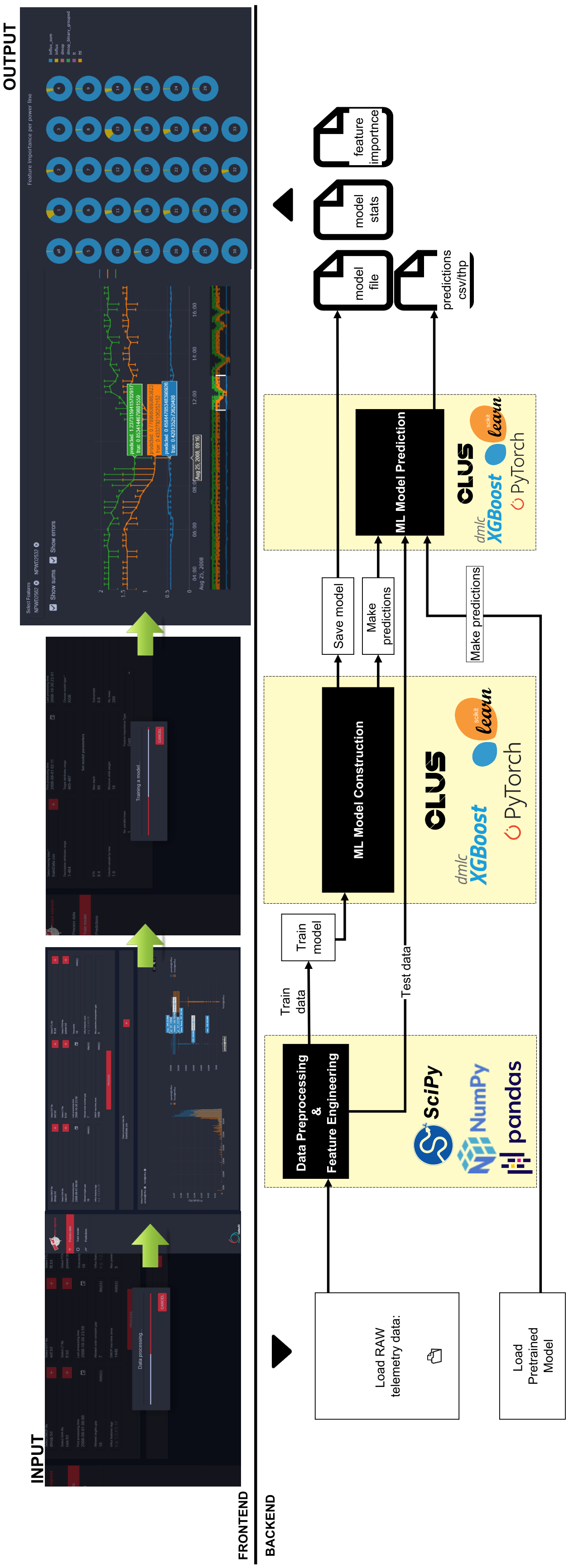

Figure 1: Overview of GalaxAI

GalaxAI-MEX employs various evaluation strategies for estimating the performance of the learned models and the quality of the predictions. Moreover, it includes a mechanism for interpreting model outputs in terms of feature importance diagrams. These evaluation statistics together with the model predictions, are the output of GalaxAI-MEX.

2.2 GalaxAI-INTEGRAL

Input data. To determine when the spacecraft enters the Van Allen belts, we rely on the on-board IREM measurements, taken every 8 seconds. The belts entry/exit times are determined through thresholding these IREM measurements. More specifically, when the counts are above 600 electron counts per second, the spacecraft is determined to be inside the belts. The orbit of each revolution of the spacecraft is defined by 12 orbital elements. We also 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 form the basis from which we engineer features that the models use to predict entry/exit times.

Data representation and preprocessing. Since we are interested in the orbital position of the spacecraft, all time stamps are first transformed to *phase* values relative to the current revolution. The phase values range from 0 at perigee to 1 at the next perigee. For this task, we consider two data representations – *positional* and *per-revolution*. The former, positional representation, is similar to the one proposed by Finn et al. [7]. Here, the data is ordered in a series where examples describe the state of the spacecraft using the orbital elements and the IREM counts (or binary indicators whether INTEGRAL is in the belts or not). Thus one can consider two different tasks: regression (when predicting the IREM counts) or classification (when predicting the binary indicator).

Learning models. GalaxAI-INTEGRAL implements several machine learning methods: (1) k-nearest neighbor regressor (KNN), (2) random forest ensembles of regression trees (RF), (3) extreme gradient boosting ensembles of regression trees (XGB), (4) gradient boosting ensembles of regression trees with quantile loss (GB), (5) fully connected neural networks (FCNN), (6) recurrent neural networks (RNN) with gated recurrent units. For some of the methods (KNN, RF, and GB), GalaxAI-INTEGRAL employs the *scikit-learn* [16] implementations. For XGB, GalaxAI-INTEGRAL uses the *xgboost* Python library [5].

3 Interaction with data and models

GalaxAI provides The machine learning pipelines that are executable trough the toolbox are well-structured, documented, and accessible by command-line interface, albeit most-well suited for data science practitioners. Nevertheless, such usage-scenarios can create some serious non-trivial challenges when used by engineers and operators who do not have prior experience working with ML-based frameworks. Such challenges include the choice of the predictive model, choosing and setting model parameters, interpretability of the model as well as explainability of its findings. The latter two, in particular, are very important when it comes to increasing the trustworthiness and facilitating the utility of predictive models in practice, especially when working with black-box models such as neural networks.

GalaxAI addresses these challenges by employing a user-friendly GUI for executing ML pipeline(s), allowing for both visual exploratory data analysis and visualization of the model results. In particular, GalaxAI facilitates seamless execution of the ML pipelines by providing pre-selected learning methods with optimal parameters (selected based on comprehensive experimental study). Next, the interactive nature of the visualizations enables the domain experts to perform exploratory data analysis on the preprocessed data and interpret the obtained models and predictions. Moreover, GalaxAI allows for performing various 'what-if' analysis scenarios by excluding data examples and/or features.

The GalaxAI GUI supports three types of visualizations:

Exploratory Data Analysis. Within GalaxAI, we have implemented interactive diagrams (histograms and boxplots) that enable users to explore the data. More specifically, the diagrams allow users to select time-ranges for visualization as well as to select several variables at the same time.

Predictions Visualization. In terms of visualising the model output, GalaxAI implements several diagrams pertaining to visualization of the obtained predictions and visualization of the influence/importance of the descriptive features. The former involves an interactive scatterplot for visual inspection of the predicted values. The latter gives more general overview, allowing for quick assessment of the predictive analysis.

Feature Importance Visualization. GalaxAI allows for visual inspection of the feature-influence within the predictive models. More specifically, it implements a special type of interactive 'doughnut' charts and pie charts for global and local visualization of the importance of the predictive features to the predictive task at hand. These charts provide the means for better explainability of both models and predictions. Namely, a feature is important when a model relies on it for predictions. Thus, by observing the importance, one can explain, to a certain extent, the model's predictions.

4 Conclusion

Spacecraft monitoring and operation involve many challenging tasks and decisions - most often based on analysis of large volumes of complex, multimodal, and heterogeneous telemetry data. These analysis, in turn, are used for monitoring the spacecraft's health as well as short/long-term operations planning. Therefore they need to be very accurate, but more importantly, they need to provide better understanding of the spacecraft's status and support the decisions of the mission operators and engineers.

In this work, we present GalaxAI - a versatile machine learning toolbox for accurate, efficient and interpretable end-to-end data analysis of spacecraft telemetry data. It implements various machine learning pipelines that are well-structured, documented, and accessible by command-line interface useful for data science practitioners. It also offers user-friendly graphical interface for executing the underlying machine learning pipelines, and performing visual exploratory data analysis and model visualizations.

References

[1] Hyojung Ahn, Dawoon Jung, and Han-Lim Choi. Deep generative models-based anomaly detection for spacecraft control systems. *Sensors*, 20(7):1991:1–20, 2020.

[2] Redouane Boumghar, Luke Lucas, and Alessandro Donati. Machine learning in operations for the mars express orbiter. In *15th International Conference on Space Operations*, Marseille, France, 2018.

[3] Martin Breskvar, Dragi Kocev, Jurica Levatić, Aljaž Osojnik, Matej Petković, Nikola Simidjievski, Bernard Ženko, Redouane Boumghar, and Luke Lucas. Predicting thermal power consumption of the mars express satellite with machine learning. In *Proceedings of the 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, pages 88–93, 2017.

[4] Ashley Carlton, Rachel Morgan, Whitney Lohmeyer, and Kerri Cahoy. Telemetry fault-detection algorithms: Applications for spacecraft monitoring and space environment sensing. *Journal of Aerospace Information Systems*, 15(5):239–252, 2018.

[5] 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.

[6] Facebook. React. https://reactjs.org/, 2021.

[7] 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.

[8] OpenJS Foundation. Electron.js. https://www.electronjs.org/, 2021.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[10] Sara K. Ibrahim, Ayman Ahmed, M. Amal Eldin Zeidan, and Ibrahim E. Ziedan. Machine learning methods for spacecraft telemetry mining. *IEEE Transactions on Aerospace and Electronic Systems*, 55(4):1816–1827, 2019.

[11] Plotly Technologies Inc. Collaborative data science. https://plot.ly, 2015.

[12] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817 – 833, 2013.

[13] Luke Lucas and Redouane Boumghar. Machine learning for spacecraft operations support - The Mars Express Power Challenge. In *Proceedings of the Sixth International Conference on Space Mission Challenges for Information Technology, SMC-IT 2017*, pages 82–87, 2017.

[14] José-Antonio Martínez-Heras and Alessandro Donati. Enhanced telemetry monitoring with novelty detection. *AI Magazine*, 35(4):37–46, 2014.

[15] Amy McGovern and Kiri L. Wagstaff. Machine learning in space: extending our reach. *Machine Learning*, 84(3):335–340, 2011.

[16] 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.

[17] Matej Petković, Redouane Boumghar, Martin Breskvar, Sašo Džeroski, Dragi Kocev, Jurica Levatić, Luke Lucas, Aljaž Osojnik, Bernard Ženko, and Nikola Simidjievski. Machine learning for predicting thermal power consumption of the mars express spacecraft. *IEEE Aerospace and Electronic Systems Magazine*, 34(7):46–60, 2019.

[18] Matej Petković, Dragi Kocev, and Sašo Džeroski. Feature ranking for multi-target regression. *Machine Learning*, 109:1179–1204, 2020.

[19] Matej Petković, Luke Lucas, Dragi Kocev, Sašo Džeroski, Redouane Boumghar, and Nikola Simidjievski. Quantifying the effects of gyroless flying of the mars express spacecraft with machine learning. In *Proceedings of the 2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, pages 9–16, 2019.

[20] Barbara Pilastre, Loïc Boussouf, Stéphane D'Escrivan, and Jean-Yves Tourneret. Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning. *Signal Processing*, 168:107320, 2020.

[21] Cunsong Wang, Ningyun Lu, Yuehua Cheng, and Bin Jiang. A telemetry data based diagnostic health monitoring strategy for in-orbit spacecrafts with component degradation. *Advances in Mechanical Engineering*, 11(4):1–14, 2019.

[22] Takehisa Yairi, Naoya Takeishi, Tetsuo Oda, Yuta Nakajima, Naoki Nishimura, and Noboru Takata. A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 53(3):1384–1401, 2017.