DOI: 10.13009/EUCASS2023-851
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

Applying Machine Learning to Routine Satellite Ground Segment Operations by Means of Automated Anomaly Detection

Leonard Schlag[⋆†], *Clemens Schefels*[⋆], *and Kathrin Helmsauer*[⋆]
[⋆]*German Space Operations Center, Deutsches Zentrum für Luft- und Raumfahrt e. V., German Aerospace Center*
Münchener Straße 20, 82234 Weßling, Germany
leonard.schlag@dlr.de · clemens.schefels@dlr.de · kathrin.helmsauer@dlr.de
[†]Corresponding author

Abstract
To tackle the domain specific challenges spacecraft operations poses on anomaly detection methods, the Automated Telemetry Health Monitoring System (ATHMoS) was developed at the German Space Operations Center (GSOC) and integrated into our Visualisation and Data Analysis software (ViDA). The main challenges include the peculiarities of the telemetry data transmitted by the satellites, the limitation of resources and accuracy and usability requirements posed by the end users. The ATHMoS was designed with these challenges in mind and uses sets of generic statistical properties to derive an explainable anomaly probability.

1. Introduction

Once a spacecraft is in orbit, engineers in charge of its maintenance and daily operation depend on telemetry data to track and analyse the satellite's status and health. As modern spacecraft generate more and more data, manually handling, inspecting, and interpreting these data becomes increasingly difficult. To assist engineers with these challenges, an anomaly detection tool, named the Automated Telemetry Health Monitoring System (ATHMoS), was developed at the German Space Operations Center (GSOC).

In this paper, we first give an overview of relevant publications that apply anomaly detection to the domain of satellite operations in Section 2.

In Section 3, we describe the unique challenges and requirements the domain of spacecraft operations poses to an anomaly detection system. These do not only apply to the algorithm and the resulting software tool itself, but also to the specification of the use-cases and to the usage of the final tool. They include, for example, the irregularity of the transmitted telemetry data, limitation of resources, and intuitive interaction with the tool for the end user.

Relating to the challenges summarised in Section 3, solutions developed at the GSOC as well as the reasoning behind the selected algorithm, workflow, and user interaction patterns are detailed in Section 4. The central steps within the ATHMoS as well as key features of our Visualisation and Data Analysis software (ViDA) are elaborated. Both of these systems have reached an operational level and are used in routine operations at our satellite control center.

Lastly, a short summary of the paper as well as an outlook on future work is provided in Section 5.

2. Related Work

Due to its capabilities to detect novelties in various domains, Machine learning (ML) has made its way into many areas of research and applications, ranging from fraud or malware detection to medical image analysis [2]. It is no surprise that it has also made its way into the domain of satellite operations for anomaly detection as well as other areas such as mission planning or image processing [5].

With regard to anomaly detection, various methodologies are currently being researched and applied to satellite telemetry with the goal of benefiting operations on ground. The Centre National d'Études Spatiales (CNES) is employing a One-Class Support Vector Machine (OC-SVN) in their New Operational SofTwaRe for Automatic Detection of Anomalies based on Machine-learning and Unsupervised feature Selection (NOSTRADAMUS) tool [3]. Their proposed anomaly score measures the distance from the day's feature vector to the decision frontier determined by the OC-SVN for the normal training data set. The daily anomaly scores then get displayed in their NOSTRADAMUS

DOI: 10.13009/EUCASS2023-851

Human Machine Interface. In addition, days with high anomaly scores are manually investigated and tracked in a history database. Moreover, the authors describe the architecture of their experimental on-board anomaly detection algorithm called Space Experiment for Satellite Artificial intelligence Monitoring (SESAM), which is based on NOSTRADAMUS. After an on-ground training, the model gets uploaded onto the satellite and performs detections on-board. SESAM is currently tested for a limited number of parameters on ESA's OPS-SAT.

To detect anomalies, EUMETSAT applies a distance-based k Nearest Neighbours (k-NN) approach and displays the results in CHART, which is both their telemetry database and front-end web GUI for visualising and processing telemetry data [4]. To explain the results to the engineers, the generated Outlier Detection Report also includes certain 2D-scatter plots of two out of the four features (minimum, mean, maximum, and standard deviation per orbit) used for the algorithm. It is currently implemented to detect anomalies of the MetOp mission.

Investigations into multivariate anomaly detection by Tariq et al. propose a multivariate convolutional Long Short-Term Memory (LSTM) network with mixtures of probabilistic principal component analysis to be trained per subsystem, which is also being used as a prototype for anomaly detection for KOMPSAT-2 [11].

Other approaches for multivariate anomaly detections, developed by B. Pilastre and supported by CNES and Airbus Defence & Space, use a sparse decomposition of a signal into a dictionary as part of a new algorithm named ADDICT [7]. It allows for including both continuous as well as discrete signals in its learned model.

Airbus Defence & Space also explores options for an ML-based Failure Detection, Isolation and Recovery (FDIR) system, including systems based on autoencoders and feed-forward models [1].

3. Challenges and Requirements

In this section, we first describe the challenges and requirements we observed and considered during the research and development of our anomaly detection system ATHMoS. Their combination is unique to the domain of spacecraft operations and reflects the experience we gained from mission operations at the GSOC.

3.1 Data Peculiarities

The time-series data, which serves as the main input to our anomaly detection system, is made up of raw satellite telemetry data. It possesses a combination of properties that increase the difficulty in applying standard anomaly detection approaches out of the box.

The data itself stems from sensors and evaluations on board the satellite. It describes attributes such as the temperatures of satellite components, the load level of the batteries, the pressure within fuel tanks, or the spacecraft's position and velocity. The telemetry data also includes the on- or off-status of, e.g., instruments such as the satellite's payloads.

The large variety of attributes leads to many different behaviours of the associated time-series data, see Fig. 1 for examples. As we want to be able to analyse each telemetry parameter individually, an approach for detecting anomalies is required to handle many different behaviours while still showing excellent results in terms of detections.

For past and current missions, no all encompassing and sufficiently labelled dataset is available. This limitation reduces the list of possible methods that can be used for anomaly detection as supervised machine learning methods can not be applied.

Furthermore, the sampling rates and behaviours are not homogeneous even for individual telemetry parameters. Causes for the varying sampling rates are primarily ground station passes of satellites in a low earth orbit or critical operations. During these events, the sampling rate is typically increased. The behaviour of the time-series itself can also vary or evolve throughout the mission. One of the main and expected reasons for changes in a telemetry parameter's behaviour is the degradation of the satellite's components and hardware. It can lead to, e.g., an increase of noise picked up by the sensors as the mission progresses or slow trends in the data due to, e.g., the degradation of the satellite's batteries. These trends are considered nominal in most cases and should not be flagged as an urgent problem by an anomaly detection system.

3.2 Computational Limitations

Satellite telemetry data is often of confidential nature, due to either the scope of the satellite's mission or the owner of the satellite. As a result, restrictions apply when it comes to working with the data. In the context of applying machine learning methods, one of the greatest restrictions is that we can not use computational resources available via cloud computing when working with classified or confidential data. Therefore, the methods we apply to the data need to run on hardware available at and controlled by the GSOC.

DOI: 10.13009/EUCASS2023-851

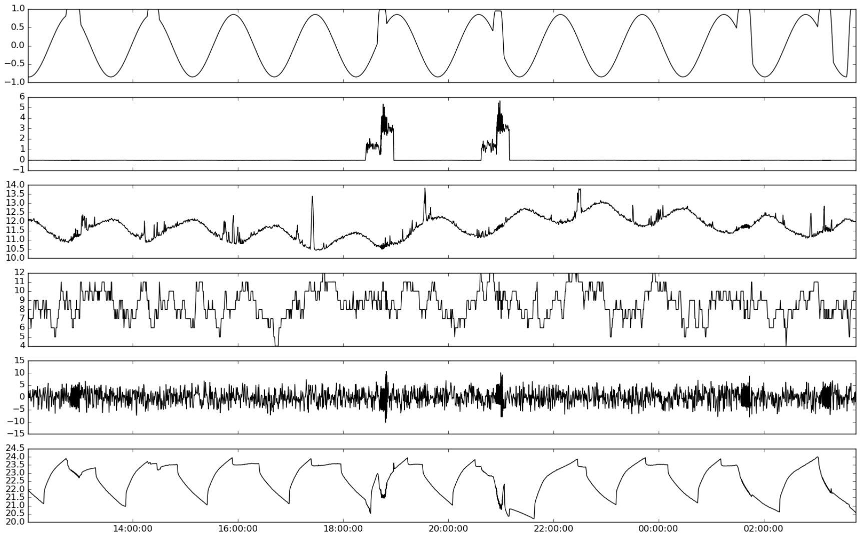

Figure 1: Examples of the nominal behaviour of six different telemetry parameters over the span of 16 hours. The parameters cover a variety of subsystems and show, e.g., temperatures, and the satellites angle with respect to the sun.

3.3 Accuracy Requirements

Modern satellites, such as the GRACE Follow-On satellites operated by the GSOC, define around 80 000 unique telemetry parameters. They allow for assessing the satellites state and enable subsystem engineers to investigate and solve possible issues that may arise. While not all of these parameters are continuously transmitted and recorded, the goal of our anomaly detection system is to automatically monitor well over a thousand parameters per mission.

Due to the sheer amount of parameters, the false positive rate of the approach used for detecting anomalies plays a vital role. While, e.g., a rate of one false positive detection per year for each parameter may seem low, a method with this false positive rate would flag around 30 false detections each day when applied to 10 000 parameters. Not only would the process of sieving through false detections be time consuming and therefore costly, it would also reduce the trust engineers have in the anomaly detection system.

In summary, a false positive rate very close to zero is necessary for an anomaly detection system in the domain of spacecraft operations. It should, however, not come at a disproportionate cost with respect to the detection rate of true anomalies.

3.4 Algorithm Properties

As mentioned in Section 3.3, the goal is to apply the anomaly detection method to thousands of telemetry parameters per mission. With this amount of different telemetry parameters, manually adjusting the algorithm for each parameter individually is not feasible. Therefore, an algorithm used for flagging anomalies has to work out of the box for almost all parameters and their behaviours. At the same time, it still needs to be easily configurable in case parameters do need special handling or the algorithm is applied to different domains.

3.5 Use Interface Requirements

In general, a User Interface (UI) needs to be intuitive, so that it is easy for users to understand and navigate without requiring extensive training or documentation. Moreover, it must feel comfortable to use such that the users willingly integrate the software into their daily routine. However, for our domain, we encountered several additional requirements that are discussed here.

One reason for using ML-based anomaly detection is the large amount of parameters we want to find anomalies in. For a human subsystem engineer, manually analysing each and every parameter would be very time consuming or even impossible. An ML based anomaly detection system, on the other hand, can provide results in a matter of seconds.

DOI: 10.13009/EUCASS2023-851

However, the interpretation and the resulting action still needs to be performed by the subsystem engineers. Therefore, the requirement on the UI is a display of detected anomalies that is fast to grasp and easy and quick to interpret.

In addition to the large quantity of parameters, the data volume of each single parameter also brings its own challenges. At the GSOC, we operate satellite missions with lifetimes of over 10 years. One parameter of these missions can hold up over 9 000 000 000 data points. This leads to three requirements: first, the data needs to be quickly loaded into the UI from the database. Long waiting times would be frustrating and less comfortable for users. Second, the data needs to be quickly visualised, even if a user wants to plot the whole lifespan of a long lasting mission. Plotting the whole parameter's lifespan with its anomalies can add valuable support to the system engineers as they can, for example, recognise similar anomalies or behaviours in historic data and act accordingly. Third, the plotted data must be reduced and compressed while still maintaining its visual characteristics. This requirement is introduced due to computer monitors only having a limited capability of plotting data points depending on their resolution. It is important for users to get a good visual impression of the data, even in longtime plots, and to be able to recognise patterns and the telemetry's characteristic behaviours even in the reduced and compressed data.

Only displaying the anomaly detection results is not sufficient to draw the right conclusions for safe satellite operations. System engineers need to understand why data was marked as anomalous. Therefore, the UI has to include functionalities providing explainability of the results, thereby enabling the system engineers to make the right decisions and ensure safe operations.

With a UI, standard analysis tools are provided to the users. Sometimes this is not enough to investigate the cause of an anomaly or a failure. Therefore, an anomaly detection system should provide an API to enable custom analysis. It should be very generic and allow the utilisation of a wide scope of data analysis method.

The last requirements come from the team that manages the software. From their perspective, the software needs to be easy to install, maintain, and operate to keep the mission costs low and operations economic.

4. Anomaly Detection and Analysis Approach

After having described the main challenges and requirements in Section 3, we detail how the approach developed at the GSOC and realised in the ATHMoS and ViDA handles the aforementioned difficulties. Finer details of the ATHMoS, the algorithm at its core, its benchmarking and its adjustment to operational usage can be found in O'Meara et al. [6], Schlag et al. [10], and Schefels et al. [8].

4.1 Algorithm Selection

At the core of any anomaly detection system lies an algorithm scoring new data, in our case satellite telemetry, based on learned behaviour from past data. Depending on the computed outlier score, new telemetry samples are flagged as anomalous. We want to briefly outline the ATHMoS before pinpointing how it tackles the various challenges.

The ATHMoS typically uses one year of past data to generate a trained model $\mathcal{M}$ as outlined in Alg. 1. We accomplish this by splitting our input data into short intervals, each spanning 1.5 hours. After removing intervals containing faulty data, we compute feature vectors containing descriptive statistics for each interval. As our input is not labelled and therefore not guaranteed to only contain nominal data, we clean the feature vectors using a clustering algorithm. Based on each feature vector's k-nearest neighbourhood, a probabilistic outlier score using the intrinsic dimension is calculated [12]. The k-nearest neighbourhood, the intrinsic dimension and the probabilistic intrinsic dimension outlier score make up our trained model $\mathcal{M}$.

New telemetry data is tested against the computed model $\mathcal{M}$ by applying the OPVID [6] algorithm, resulting in scores between 0 and 100 percent. This score roughly describes how similar the descriptive feature vector computed for the new telemetry data is to the ones used in the trained model. A threshold is then applied to decide whether the new data should be flagged as anomalous, in our case with additional steps to improve the false positive rate (see Section 4.3). The model itself is updated on a weekly basis only considering past data which was not flagged as an anomaly.

The algorithm sketched in Alg. 1 already considers many of the issues described in Section 3. The large variety of behaviours presented in Section 3.1 is handled in multiple ways. The model containing the past behaviours represents one year of data which includes both seasonal behaviour as well as slowly evolving trends. The weekly update of the model ensures gradual changes in the telemetry's behaviour are continuously learned and included in the model.

Apart from one parameter showing different behaviours, the vast variety of behaviours across parameters are also considered. The distinction between continuous parameters, such as temperatures, and discrete parameters, such as binary on/off flags in lines 3 and 5 of Alg. 1, ensure the compatibility with discrete parameters. In addition, the statistical features making up the feature vectors computed in line 5 of Alg. 1 were engineered to work with many different behaviour types and optimised in an exhaustive test period. The successfully reached goal of this optimisation was a

DOI: 10.13009/EUCASS2023-851

Data: Historic training time-series data for one parameter ($\approx$ 1 year)
Result: Trained model $\mathcal{M}$ based on historic data

1 Decompose the time-series data into overlapping intervals I_i. Each interval has a size of 1.5 hours and we use a sliding window step size of 30 minutes;
2 Remove intervals I_i containing gaps larger than a configurable length and apply linear interpolation to the remaining intervals;
3 Determine the parameter type, either discrete or continuous;
4 **for** *each time-series interval* I_i **do**
5   Based on the determined parameter type, calculate the feature vector f_i composed of various statistics as described in Schlag et al. [10];
6 **end**
7 Use clustering methods to remove feature vectors $\overrightarrow{f}_i$ that represent possible anomalies, resulting in a set of nominal feature vectors $\tilde{F}$;
8 Compute a k-nearest neighbourhood model KNN_h for $\tilde{F}$ with a sufficiently large k;
9 **for** *each* $\overrightarrow{f}_i \in \tilde{F}$ **do**
10   Calculate the feature vector's intrinsic dimension id_i;
11   Calculate the Probabilistic Intrinsic Dimension Outlier Score $pidos_i$;
12 **end**
13 $\mathcal{M} = \{KNN_h,\ ID_h,\ PIDOS_h\}$

Algorithm 1: Training of the historic model $\mathcal{M}$ required for computing the outlier score, see Schlag et al. [9].

low false positive rate while still correctly flagging known anomalies. As the features are based on statistical attributes of the data, they are also inherently understood by the users of the system, thereby increasing the explainability of our approach.

The use of the intrinsic dimension computed in line 10 of Alg. 1 based on Brünken et al. [12] provides additional advantages. Not only does the OPVID algorithm using the intrinsic dimension show a good separation between anomalies and nominal data, it also allows for large feature vector dimensions without significant loss of quality [10]. These attributes make it easy to configure the algorithm, as new features can be added to the feature vector without any issues. Additional features could, e.g., include unforeseen behaviours or properties needing to be considered in the anomaly detection. Defining a whole new set of features is also easily possible and can extend our approach to different domains or applications. In summary, the use of the intrinsic dimension together with the distinction between parameter types and optimised feature vector composition solve many of the requirements noted in Sections 3.3 and 3.4.

In order to reach the goal of working for thousands of parameters, the computational performance of the approach has to be taken into account (see Section 3.2). It has to be performant enough to work on local hardware available at the GSOC. Throughout the development of the ATHMoS, this limitation was kept in mind. Both the weekly training of the model and the daily testing of new data is feasible in a short amount of time on hardware provisioned at the GSOC, even for large amounts of parameters. Testing new data spanning a whole day for a single parameter is possible within seconds and easily parallelisable. While retraining takes a bit longer, it can be completed in a few hours for over a thousand parameters and parallelised as well.

As the reduction of the false positive rate is one of the key challenges we are faced with when applying the ATHMoS to many different satellite telemetry parameters, two additional steps were taken to reduce this rate even further. The statistical features composing the feature vectors are not only computed based on the raw telemetry data, but also include features derived from smoothed data. The additional smoothing reduces the false positive rate due to our data often containing noise or behaving in a noisy manner [10]. Features extracted from smoothed data show less variance than the same features based on unsmoothed data and are therefore less prone to false positives. The second additional step taken to reduce the false positive rate was benchmarking our algorithms against other methods using publicly available as well as synthetic benchmarks [10]. The results of these benchmarks were used to tune the algorithm parameters.

4.2 Data Labelling

The issue of unlabelled data described in Section 3.1 is tackled in two ways. While Alg. 1 is unsupervised by design, the input data is cleaned during the initial training using a clustering algorithm (see Alg. 1 line 7). Thereby, we

DOI: 10.13009/EUCASS2023-851

prune behaviours in the training data that appear in less than one percent of the segmented time intervals. By doing so, possible anomalies or outliers are not included in our trained model. In addition, expert knowledge can easily be considered by the algorithm. If a new behaviour appears, it should at first be flagged as an anomaly. Should our users, the subsystem engineers, decide that the new behaviour is nominal, the intervals containing the new behaviour can be relabelled to *nominal*. The next time the model is retrained, the relabelled intervals are considered. This way, the new nominal behaviours are included in the model and will not be falsely flagged as anomalies in the future.

4.3 Post-processing

As mentioned in Section 4.1, additional post-processing steps are applied to intervals with anomaly scores surpassing a threshold of 90 percent. The goals of these steps are both the reduction of false positives as well as supplying additional context in case of slowly changing behaviours.

We exploit the overlapping nature of the time intervals we segment both the training as well as the test data into (see Alg. 1, line 1). With windows of length 90 minutes and steps of 30 minutes, three consecutive windows share 30 minutes of data. In other words, each 30 minute interval in our test data is included in three separate windows. We therefore label a window as a high priority anomaly detection if the anomaly score computed for all three consecutive windows breach the threshold. This additional measure can be interpreted as the anomaly needing to be significant enough to influence all three windows it is a part of. This additional labelling is visualised in Fig. 2.

As a safeguard for mislabelling trends as anomalies needing urgent attention, in our case as a high priority anomaly, we implemented additional post-processing based on the output of a secondary model. This model, which we denote by $\mathcal{M}_r$, is trained with the last month of data and retrained on a daily basis. Compared to the model which is trained on a year of data, which we denote by $\mathcal{M}_h$ in Table 1, the model $\mathcal{M}_r$ captures the most recent behaviour appearing in a parameters telemetry. New data is therefore less likely to be marked as a high priority anomaly when tested against this is model. In the ATHMoS, we test data against both $\mathcal{M}_h$ capturing a full year of a parameters behaviour and $\mathcal{M}_r$ capturing its most recent behaviour. The results are merged according to Table 1. An example for a parameter describing an on-board temperature evaluated with the ATHMoS is given in Fig. 3.

4.4 Result Visualisation and Presentation

The requirements identified in Section 3.5 have been heavily considered in the development of our *Visualisation and Data Analysis* software ViDA. In addition to many other features, this web application is used to visualise and present

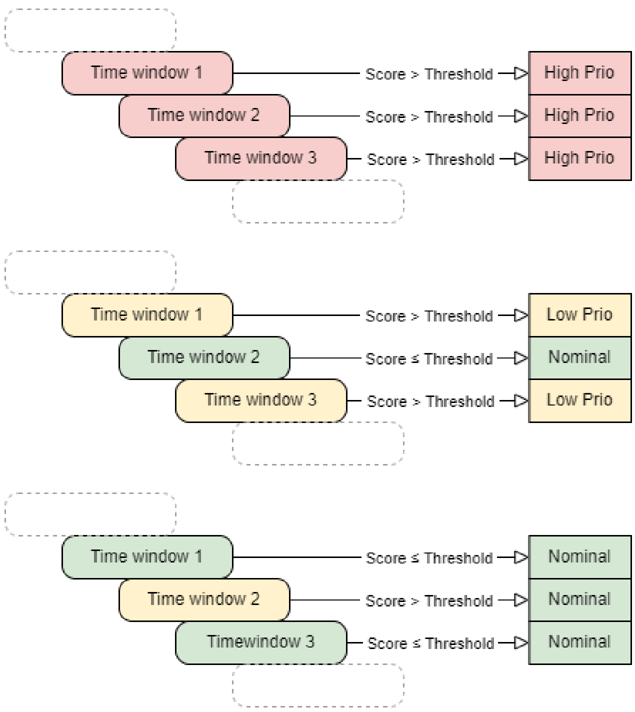

Figure 2: Labelling of overlapping time windows, see Schefels et al. [8].

DOI: 10.13009/EUCASS2023-851

Table 1: Overall labelling schema for detections taking into account the results w.r.t. $\mathcal{M}_h$ and $\mathcal{M}_r$ [9].

$\mathcal{M}_h$ \ $\mathcal{M}_r$	High Priority	Low Priority	Nominal
High Priority	High Priority	Low Priority	Trending Detection
Low Priority	Low Priority	Low Priority	Low Priority
Nominal	Low Priority	Nominal	Nominal

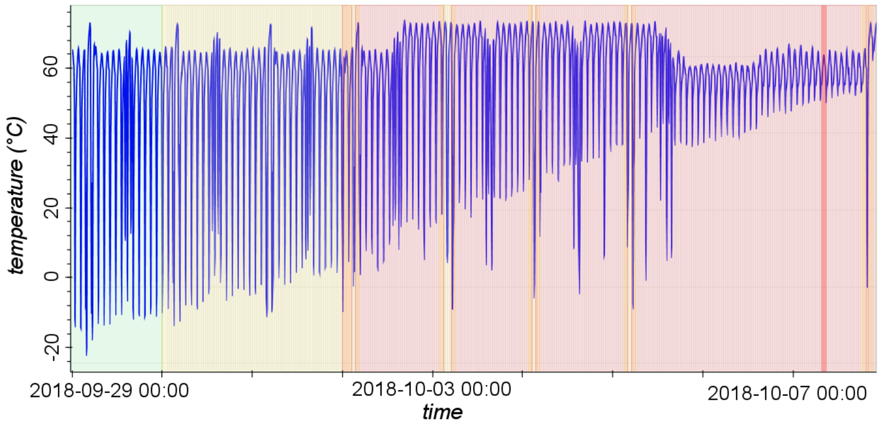

Figure 3: Telemetry parameter going from nominal, here marked in green, to trending, coloured in yellow. Once flagged as a high priority detection by the ATHMoS, it is highlighted in red [9].

the results of the ATHMoS to the users. To give the user a modern and up-to-date front-end, we use the common Angular framework. The framework ensures that the majority of our users are familiar with the UI elements. For the back-end, we use a state of the art micro-service based architecture to ingest, pre-process, and access the telemetry data [8].

After logging into ViDA, the user is presented with a compact overview of each satellite's status including the range of its available data and the detected anomalies as depicted in Fig. 4. Subsystem engineers can get a first impression of their monitored satellites' status and detected anomalies within seconds, even if the satellites transmit a large amount of parameters. We thereby satisfy the requirements of the representation of anomalies defined in Section 3.5.

For a detailed inspection, the subsystem engineers can select parameters belonging to various system groups or single parameters and plot the telemetry data, see Fig. 5. One strength of ViDA is its capability to plot parameters with many data points in a matter of seconds. To achieve this, we use a database tailored to time-series data in combination with a downsampling algorithm. The downsampling algorithm is able to reduce the data needing to be displayed while still maintaining the visual characteristic of the telemetry data. As a result, the system engineers are able to plot the whole mission lifespan of a parameter in close to real-time. We also include the detected anomalies as red bars in these plots. The different shades of red mark the different types of anomalies (see Section 4.3 and Fig. 2). In summary, the challenges of displaying parameters covering billions of data points quickly, compressing the data while maintaining its characteristics, and visualising detected anomalies in the context of the whole mission are tackled.

In our continuous development of ViDA, we plan to include additional information related to the parameters from various sources in the plots such as executed telecommands, on-board events, or manually entered anomaly reports. Additionally, dependency graphs of parameters and feature vector visualisations will provide the users with more explainability of the detected anomalies.

To allow for a custom analysis outside of ViDA, we also provide a Python library called MiDA (*Mission Data Access*). Using this library, users can download and analyse all available data in their personal analysis workspace, e.g., in Jupyter notebooks.

5. Summary and Outlook

This paper described the main challenges the domain of satellite operations poses to a machine learning based anomaly detection system and how they were tackled at the GSOC. These challenges include the large amount of different

DOI: 10.13009/EUCASS2023-851

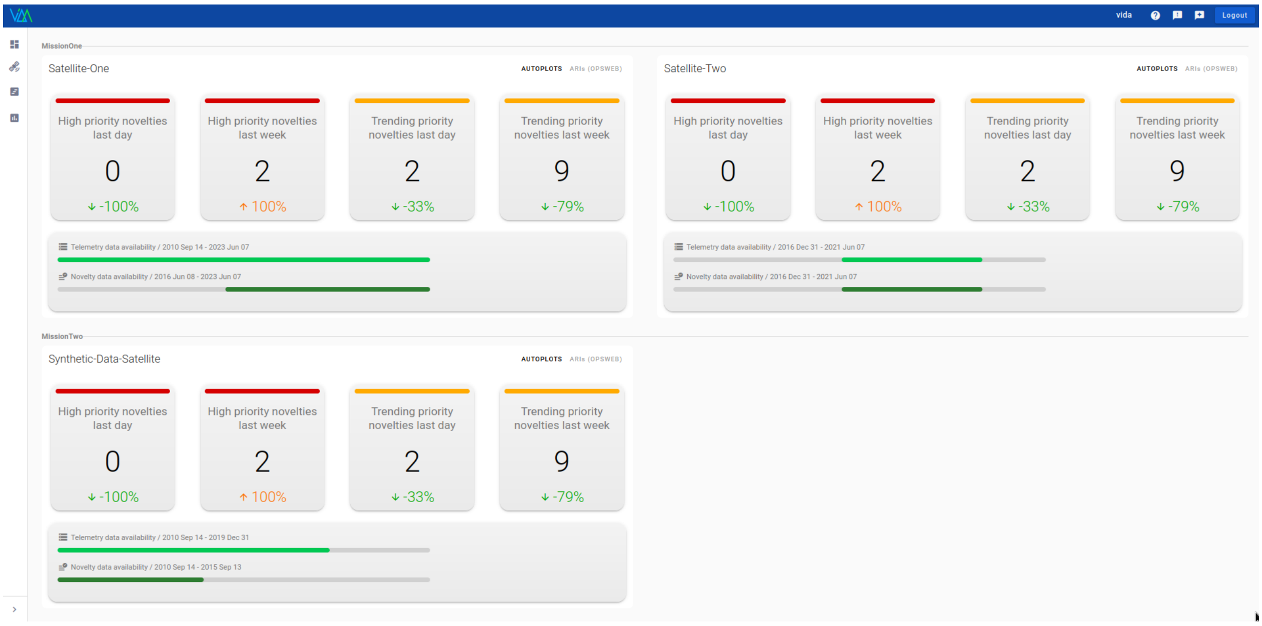

Figure 4: The ViDA dashboard: Quick overview about the satellites' status including available data and detected anomalies.

telemetry parameters and behaviours, the confidentiality of the data, and the requirements posed to the UI displaying the telemetry data and anomaly detections. We provided insights into how each of these challenges was solved within our anomaly detection system ATHMoS as well as our visualisation front-end ViDA.

To extend our system in the future, feedback gathered from its operational use will be considered and utilised. As a first enhancement, we want to improve both the explainability of new detections as well as the correct handling of new or falsely flagged behaviours by means of classification. If users relabel an incorrect detection as nominal or provide further labels to detections and enrich them with context, these labels based on the user's expert knowledge could be used to classify new detections. For new detections, the goal is to automatically compare them to these labelled samples and, in case of sufficient similarity, assign the matching label. In doing so, we would both decrease the time it takes to learn new behaviour and provide additional context to users should a similar anomaly have already

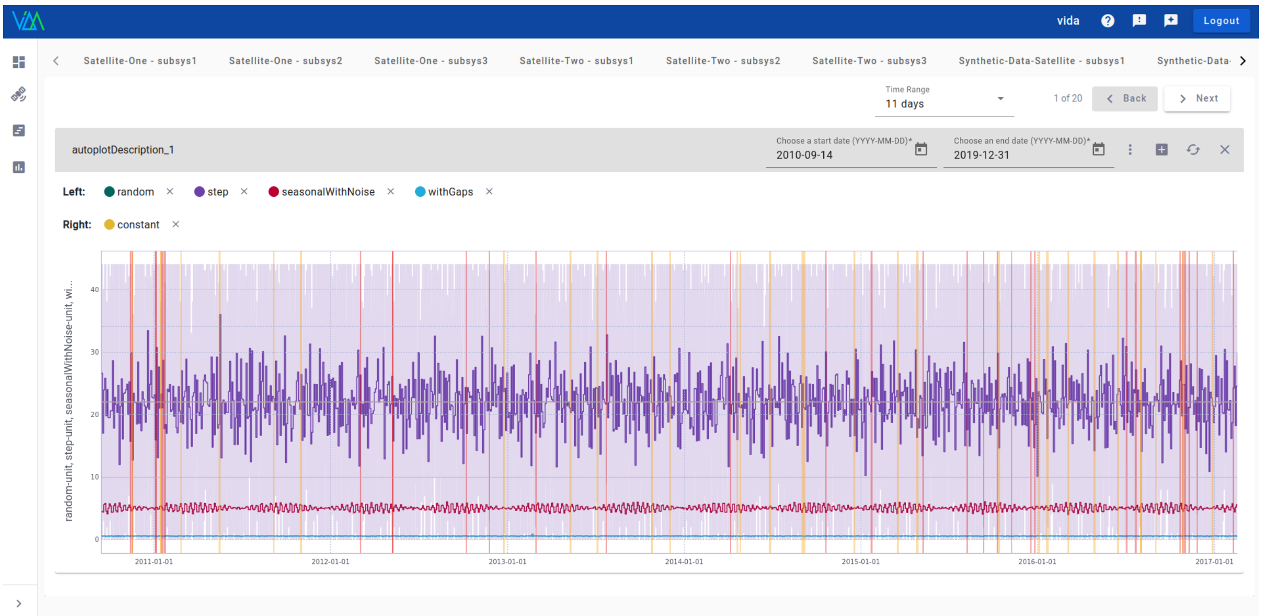

Figure 5: Plot of five telemetry parameters with detected anomalies (red bands) in ViDA.

DOI: 10.13009/EUCASS2023-851

occurred in the past.

We will also continue to investigate further applications of our core algorithm used in the ATHMoS. A natural fit is applying the method to real-time telemetry data, both on ground and on board. Here, we can build on the experience gained from the current system and focus on extending its functionalities to deal with the additional requirements introduced by real-time data. A first prototype for detecting anomalies in real-time data using our approach has already been build and shown promising results. Including pre- and post-processing, the computations required to derive the anomaly score were performed in fractions of a second and anomalies introduced in a stream of synthetically generated data were detected.

References

[1] Filippo Ales, Alisa Krstova, Thomas Chabot, Max Ghiglione, Mario Castro de Lera, Florian Hegwein, Andreas Koch, Carlos Hervas Garcia, Prem Harikrishnan, Maen Mallah, Rashid Ali, Michael Rothec, and Laurent Hili. Spacecraft Failure Detection, Isolation and Recovery using Artificial Intelligence: A Study forImplementation on the Edge. In *17th International Conference on Space Operations*, 2023.

[2] Raghavendra Chalapathy and Sanjay Chawla. Deep Learning for Anomaly Detection: A Survey. https://arxiv.org/abs/1901.03407, 2019. arXiv.

[3] Pauline Delande, Pierre-Baptiste Lambert, Mouadh Bouayad, Mathias Zaroubian, Audric Baron, and Eva Jalabert. Ai for Satellite Anomaly Detection: On-Ground Operational Feedback and Development of On-Board Experiments. In *73rd International Astronautical Congress (IAC)*, 2022.

[4] P. L. Losco, A. De Vincenzis, J. Pergoli, and R. Dyer. From Theory to Practice: Operational Implementation of Telemetry Outlier Detection at EUMETSAT. In *16th International Conference on Space Operations*, 2021.

[5] Pablo Miralles, Antonio Scannapieco, Nitya Jagadam, Prerna Baranwal, Bhavin Faldu, Ruchita Abhang, Sahil Bhatia, Sebastien Bonnart, Ishita Bhatnagar, Pallavi Prasad, Héctor Ortega-González, Harrish Joseph, Harshal More, Sondes Morchedi, Aman Panda, Marco Di Fraia, Daniel Wischert, and Daria Stepanova. Machine Learning in Earth Observation Operations: A review. In *72rd International Astronautical Congress (IAC)*, 2021.

[6] Corey O'Meara, Leonard Schlag, Luisa Faltenbacher, and Martin Wickler. ATHMoS: Automated Telemetry Health Monitoring System at GSOC using Outlier Detection and Supervised Machine Learning. In *SpaceOps 2016 Conference*. American Institute of Aeronautics and Astronautics, 2016.

[7] Barbara Pilastre, Jean-Yves Tourneret, Stephane D'Escrivan, and Loic Boussouf. Multivariate Anomaly Detection in Mixed Telemetry time-series Using A Sparse Decomposition. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 430–434. IEEE, 2019.

[8] Clemens Schefels, Leonard Schlag, Agnese Del Moro, Kathrin Helmsauer, Tobias Lesch, and Tobias Göttfert. Bringing a Machine Learning Based Novelty Detection Software Tool from Research to Production. In *17th International Conference on Space Operations*, 2023.

[9] Leonard Schlag, Matthias Dauth, and Armin Braun. The GSOC Satellite Telemetry Framework. In *Deutscher Luft- und Raumfahrtkongress 2019*, 2019.

[10] Leonard Schlag, Corey O'Meara, and Martin Wickler. Numerical Analysis of Automated Anomaly Detection Algorithms for Satellite Telemetry. In *15th International Conference on Space Operations*, 2018.

[11] Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S. Woo. Detecting Anomalies in Space using Multivariate Convolutional LSTM with Mixtures of Probabilistic PCA. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019.

[12] Jonathan von Brünken, Michael E. Houle, and Arthur Zimek. Intrinsic Dimensional Outlier Detection in High-Dimensional Data. *NII Technical Reports*, 2015.