17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 277

ENABLING AI APPLICATIONS FOR SPACE OPERATIONS THROUGH A MULTI-MISSIONS DEVOPS PLATFORM

Pablo Beltrami [a], Nieves Salor Moral [a], Mads Hedegaard [b], Jonas Hansen [b], Nicola Policella [c], Rui Santos [d], Evridiki Ntagiou [d], Francois Trifin [d]

[a] *RHEA Group, Robert-Bosch-Str 7, 64295 Darmstadt, Germany, p.beltrami@rheagroup.com*
[b] *Airbus Defence and Space GmbH, Airbus Allee 1, 28199 Bremen, Germany*
[c] *Solenix Deutschland GmbH, Spreestrasse 3, 64295 Darmstadt, Germany*
[d] *ESA/ESOC, Robert Bosch Str7, 64295 Darmstadt, Germany*

Abstract

The AI4OPS project aims at the definition and implementation of a platform for developing and supporting AI-based applications targeting the automation of space operations and demonstrating it with a set of novel proof of Concept AI Applications. This platform enables its users to build, train, and deploy ML models of AI applications, supporting mission operations. Additionally, the platform can also be used for general data engineering and software development purposes, allowing easy interface to data from ongoing or past space missions. The goal is the provision of a "platform as a service" (PaaS) available to internal ESOC projects and to data and software developments driven by ESA affiliates, contractors or other 3rd party collaborators.

Companies, institutions and researchers may be hindered in the use of AI methods and applications due to the data access and lacking infrastructure, which typically cannot be met due to budget and knowledge limits. The common platform aims to reduce this gap with the mentioned PaaS approach, so users can focus on the logic of the AI models and not on setup, deployment and infrastructure topics, in addition to ease the process of accessing data.

For Mission Operations personnel to adopt, trust and start using the developed AInabler platform, proofs of concepts and demonstrators were required. So, three AI applications were implemented after the development of the core platform, guidelines and strategies to be followed by projects. The applications, covering use cases identified in mission operations at ESOC, allowed the demonstration and validation of the platform capabilities, using real mission data available at ESOC.

This paper presents the results of the AI4OPS activity including the platform design, initial use cases, applications and achieved results. Being just a piece of a bigger picture, the paper also shows how users interact with the platform, their impressions to be considered for the ongoing ESOC AI Roadmap implementation in operations and plans for future improvements and extensions.

Keywords: Artificial Intelligence, Machine Learning, operations support tools, AI Roadmap, ESOC.

1 Introduction

Artificial Intelligence/Machine Learning (AI/ML) technologies have reached a state of maturity where they can help to substantially increase the level of automation in ground segment operations. ESA/ESOC has established a multi-projects roadmap for the adoption of AI across the operation centre, which includes establishing and making available the required infrastructure, including processes and reference information. The roadmap defines a total of 15 Use cases within 5 domains identified in close collaboration with the ESOC teams responsible for current and future missions. ESOC has set the objective to support these use cases operationally by 2025 through multiple development activities for which AI4OPS is one of the first stepstones. The resulting platform named "AInabler" is intended to serve as common environment for the development and operation of these use cases in the future.

17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

2 Application of AI in operations

Space operations, despite having numerus AI application possibilities, have only few cases of AI solutions close to commercial success, with many challenges of adoption still holding back the full potential of AI/ML usage. Based on ESA classification [1], the following areas of application for AI in space operations can be categorized:

- Mission Planning Support and Optimization
- Automated Operations
- Ground Stations
- Operations for Telerobotics, Autonomous and Crewed Systems
- Pre-mission Data Extraction

The list might not be fully conclusive as other topics such as flight dynamics or mission analysis, could potentially be seen as separate topics from the above mentioned. By using AI, especially machine learning, to improve monitoring, investigations and predictions, the aim would be increasing the level of automatization and, thus, potentially get closer to autonomy for these aspects of space operations.

2.1 Purpose of the AInabler Platform

The AInabler platform presented in this paper is, at its core, a data platform. As a consequence, the platform has to allow data management through coding. Thus, the platform provides support for all standard development languages, tools, and libraries, which enables its users to build, (re-)train, and deploy machine learning (ML) models to support the domains above. It supports fast prototyping and testing, allowing fast judgment of the feasibility and possible value of a potential model, likewise, inspection and evaluation of data quality. The platform, however, is not restricted to only be used for AI related projects. It can also be used for other types of data science and engineering developments; augmenting its foreseen user base and aligning it with many other developments within ESA (especially ESOC), e.g. simulations and digital twins, which might also include ML, and the potential of coupling the legacy ESA applications to the platform, e.g. currently used mission planning tools, etc.

2.2 Developed Guidelines

To increase the use and adoption of AI in operations, having a platform is not enough because users need to understand and trust resulting AI application help them instead of substituting them. This is a long process where, in most cases, trust is achieved in an empirical manner. Hence, the AI4OPS activity also defined guidelines focused on the practical use of AI for ESA missions by:

- Defining an AI Strategy [2] for deploying Artificial Intelligence applications for space operations (at ESOC and elsewhere) and to get an overview of the challenges and solutions, when utilizing AI and ML.
- Defining an AI Information Architecture [3]for a multi-user platform, which allows for the development and deployment of applications to support space operations.
- Establishing Operational Data Acquisition Guidelines [4] addressing the best way for new projects to target AI since the start focusing on the needed skill set, considerations to plan beforehand and how to collect any required information by defining what information needs to be collected, why it needs to be collected, where the information is and how it must be made available.
- Establishing an Operational Data Access Governance [5] including initial processes and security aspects, analysis of data types, actors, and types of access in the context of AI/ML projects.

These documents are intended to support mission teams and other stakeholders get a better understanding of the domain and how it can be used and implemented in a ground segment environment.

3 Platform design and implementation

The platform design uses a specific DevOps chain, known as MLOps, which focuses on automating much of the lifecycle of AI applications and ML models, allowing the deployment of the applications, the continuous integration (CI), continuous delivery (CD), and continuous training (CT). ESA's AI roadmap [6] foresees a MLOps level 2 [7] for the platform at ESOC which represents near full automation of the CI/CD/CT, including gathering of model metadata and tracking models/pipelines based on versioning. Within the AI4OPS project, MLOps level 1 is the target level, not achieving the automated pipeline for model performance monitoring and CT, but more focusing on the initial automated CI/CD system. That said, the architecture is designed to be open and cloud ready to support Level 2 in the next stage of the platform evolution, to cover a wide range of missions and users (both internal and external).

3.1 Ainabler Platform Design

The platform is designed to be deployed in a Kubernetes cluster in the ESA cloud. It is based on microservices to leverage the scalability of the system as the number of users and/or resources may vary and an increase of demand is expected due to the envisaged adoption of AI within ESA. The structure of loosely coupled services reduce all types of dependencies and complexities around it. Services can be developed independently of each other, interact via well-defined APIs and be organized into groups each with a distinctive role and purpose within the platform. It is likewise easy to scale up applications by simply running duplications, and by distributing applications on multiple nodes, making the whole solution more robust to downtimes.

The initial design of the architecture is based on the lambda architecture framework [8], which defines three layers with the following service groups:

- The Batch layer is mainly used for batch processing of large amounts of operational data including the Model Training services.
- The Serving layer stores and serves the output of the batch and speed layer and includes the Storage, Model Inference, Monitoring and OPS Interface service components.
- The Speed layer is used for real-time processing of streaming data including the Data Stream Service and the individual Application components developed using the platform.

Fig. 1 below provides an overview of Kubernetes internal software architecture, represented by the following segments:

- Model training environment,
- storage of model artefacts, processed data and results,
- model interference for ML model serving and orchestration,
- application environment for deployed applications,
- user and container monitoring, including administrator interface,
- handling of data streaming for the applications and
- interface management between PRE-OPS and OPS LAN.

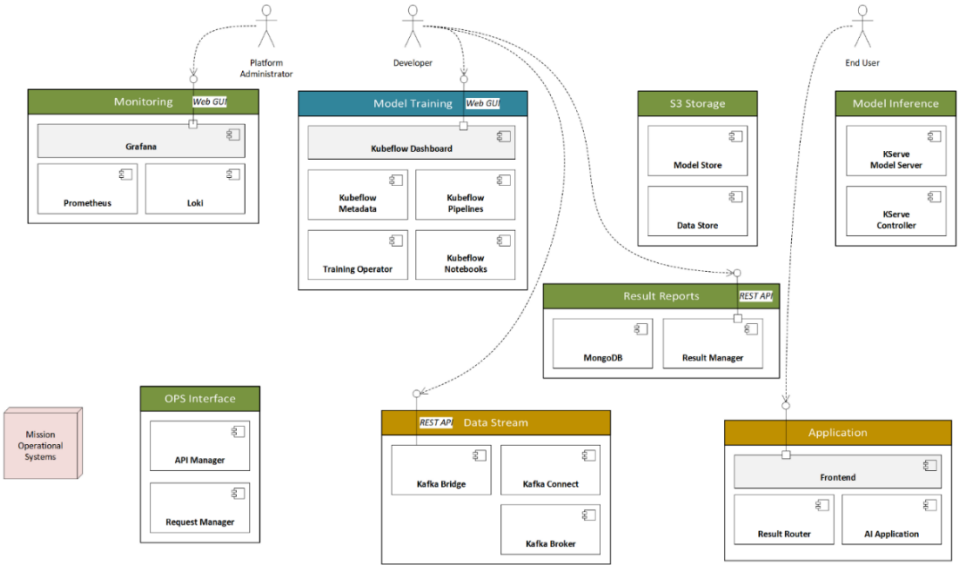

Fig. 1 Component diagram with external interfaces for different user groups. The colour coding of the group containers indicates their mapping to one of the lambda layers: batch layer is blue, speed layer is yellow and serving layer is green

In the following paragraphs, a short overview of service groups and their role within the platform is presented.

- The Model Training component group queries the OPS Interface for data, which is used for exploration, model training and model validation. It composes the whole batch layer because all batch processing is performed within this group. The output is a trained ML model or more specifically a ML pipeline which can also encompass auxiliary data processing steps next to the ML model itself. It is composed of the following sub-components:
    - The Central Dashboard (see Fig. 2) is the main GUI for the ML engineer and data scientist. It provides an overview and access to all Kubeflow components deployed on the platform. The dashboard is part of the Kubeflow platform and is based on the Node.js web application framework Express.

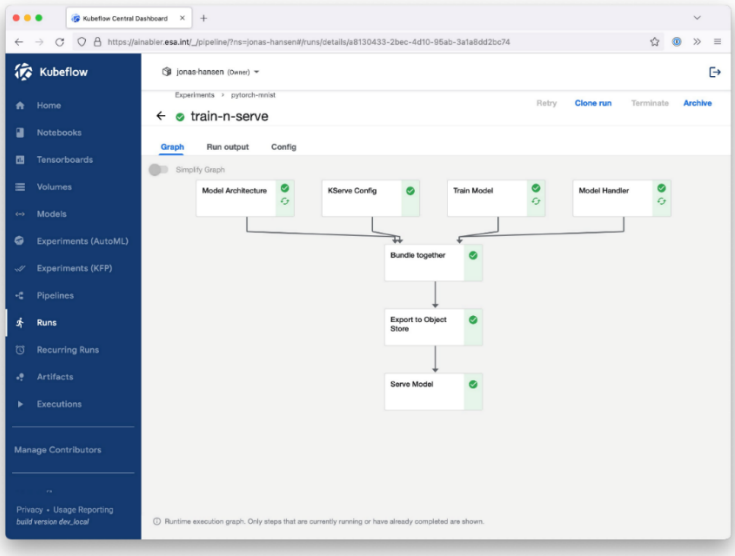

Fig. 2 Main Dashboard of Kubeflow – Pipeline run

- The ML pipeline is provided by Kubeflow Pipelines (KFP). The primary task of KFP is to enable and simplify the orchestration of ML pipelines. A pipeline is, in this context, the definition of a ML workflow, including the definition of all components and the relation between them. KFP enables the re-use of components across pipelines and, additionally, the re-use of whole pipelines to accelerate the development of new ML solutions.

- Kubeflow Notebooks (see Fig. 3) allows the developer to use the notebook development format directly in the Kubernetes cluster instead of on a local workstation. Thereby, the computational resources of the cluster can be leveraged as well as the availability of all data.

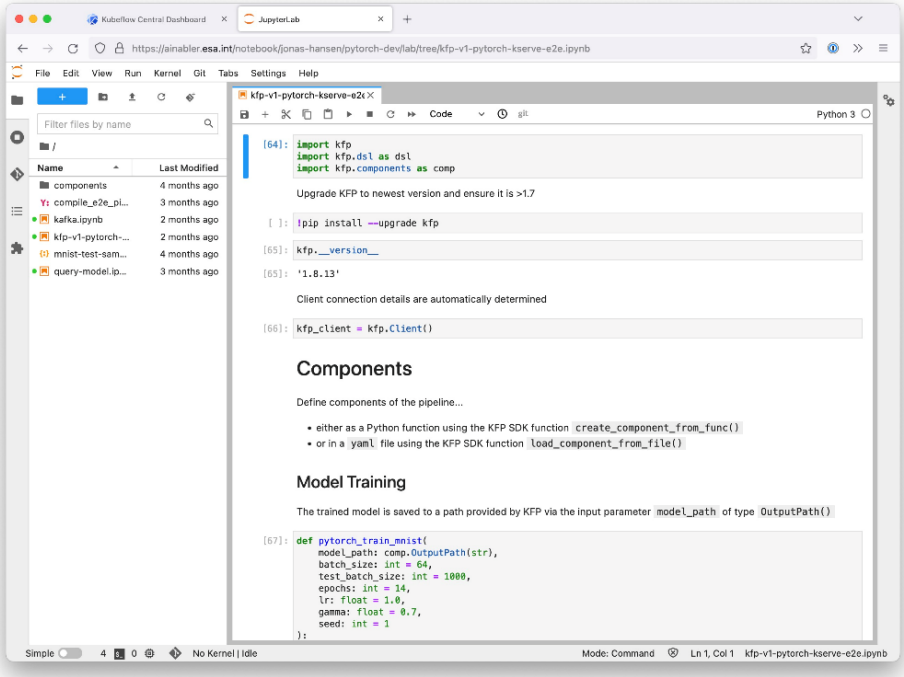

Fig. 3 AInabler Notebook

- The ML Metadata component uses the ML Metadata [9] (MLMD) library developed by Google. Each run of a ML pipelines generates metadata containing information about the various pipelines components, their executions and resulting artifacts. This metadata helps to understand and analyse all parts of the ML pipeline and their interworking. It supports the ML engineer and data scientist to answer the following questions:

    - Which dataset was the model trained on?
    - What where the hyperparameters used during training?
    - What were the metrics of the model?
    - Was any other model trained on this dataset?
    - Which pipeline run created the model?
    - Which ML framework created this model?
    - Which version of the ML framework created this model?

- The Storage component group is part of the serving layer and is responsible to persist and serve different types of data for the platform. It stores data used by the platform internally like the trained ML models/pipelines, the results of the Applications and intermediate data which has been processed in the Kubeflow Notebooks. This component group is not intended to store large amounts of data or act as a big

17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

data solution, as the Mission Operational Systems (MOS) provides dedicated storage capabilities (e.g. ARES/MUST) for these purposes. The storage component implementation is based on the MinIO library, providing a single storage system accessible via S3 API [10].

- The Model Inference component group servers all trained models/pipelines and provides a consistent interface for querying these assets and performing the inference step. The architecture and implementation are based on the KServe library [11].

- The Application component group contains all use-case specific services. While all other component groups are use-case independent and are deployed in a single manifestation, there are multiple deployments of the Application in parallel. The Application can request archive data or live data from the OPS Interface. All live data is provided as streaming data via the Data Stream component group. The incoming data can be processed and then be used to query the Model Inference. The output is any result report derived from the model output returned via an event stream. This event stream (see Data Stream component group) is persisted in the Storage component. Each AI Application is different, but in general they consist of the following steps:

    - Retrieval of archive or live data using the platform Data Manager
    - Pre-processing of data, e.g., resampling, combining and labelling data
    - Storing data in Object Storage
    - Model Training using the retrieved and processed data.
    - Storing the trained model including pre-processing pipeline in the Model Storage
    - Applying a trained model using Model Serving
      Sending the results to Result Router

- The results of an AI Application job are sent to the Result Router, which publishes them on a Kafka topic and afterwards persisted in the Result Report Database of the platform. Since different applications produce varying types of data as their results, the Result Router uses a common data model for all AI applications with a specific part which is dependent on the use case.

- The Result Report Database is based on the document database MongoDB and is the central storage location for all Result Reports generated by the AI Applications. Depending on the use-case of the AI Application, the specific format and content of a result does vary, but there is a general structure of a Result Report. It encompasses the result of the analysis performed by the AI Application specific to the trained ML model, which is relevant for the operator to process prior performing his task.

- The Data Stream component group is part of the speed layer and acts as a message bus to stream data between components with low latency and replication and fault-tolerance. It is based on the Kafka system deployed and managed by Strimzi [12] acting as a Kubernetes Operator.

- The Monitoring component is the central GUI for the user to monitor behaviour of the platform and of the deployed ML models and pipelines. It is split from the overall dataflow as it aggregates the performance metrics and logs of the other services. It also visualizes this information and can display the content of databases part of the Storage component. It consists of three major subcomponents based on Prometheus, Loki and Grafana libraries. All subcomponents are deployed as Helm charts on the Kubernetes cluster.

- The OPS Interface component group acts as a unified interface to the MOS, enabling other services in the platform to request data to be used by the AI Applications and Training models in a consistent way, independent of the underlying MOS application or API definition. This component is also responsible of providing access and querying the OPS catalogues for data and asset discoveries (including navigation between catalogues). It consists of the API Manager and the Request manager. The API Manager consumes the queues filled by the Request Manager and processes the request to adapt it to the specific MOS which has to retrieve the data information in order to avoid data silos and manage the access to different systems deployed in different networks.

17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

3.2 Infrastructure

Each microservice is implemented in a Docker container and Kubernetes manages the deployment, the scaling, the configuration and the service discovery of these containerized applications. The Kubernetes cluster consists of a number of Nodes, which correlate to virtual machines in the underlying server infrastructure. The deployment of an application can be scaled up by running multiple replicas of one container (called Pods).

3.3 CI/CD Pipeline

AInabler utilizes the CI/CD tooling of the GitLab instance deployed at ESA. This includes the Container Registry and Package Registry of GitLab. Whenever a new commit is pushed to the repository, the scripts to build and test the application are automatically triggered. These tests ensure the code changes pass all tests, guidelines and code compliance standards established by the platform. Whenever a development branch is merged into the master branch, the scripts to build, test and deploy the application are automatically triggered. The build process results in a set of Docker images which are pushed to the Container registry of GitLab and subsequently in a Helm package pushed to the Package registry of Gitlab. The deployment process of the newly available Docker images or Helm charts can either be performed manually by the developer or in a more advanced and automated scenario can be performed automatically by a GitLab Agent running in the Kubernetes cluster.

4 Data Access

In AI and ML related projects, data is key, which may be understood as AI only needing vast amount of data, but the real secret to efficient AI is data quality. However, ensuring data is of good quality, e.g. it is meaningful, properly labelled and sufficient for the training/validation processes, is a difficult task.

The increasing use of intelligent ESA systems (e. g, MUST and ARES) and MBSE across ESA missions helps providing access to operational data in a form more accessible to be used by AI methods. However, further work is still required to discover and prepare the data. In the frame of the AInabler platform development, initial steps are being taken into developing catalogues and ontologies for the data discovery. Through the definition of metadata (data that describes or summarizes data) for operational mission datasets, the platform will allow labelling and rating of the data without changing the original data sets. This will help application developers and users find the most appropriate data for their purposes and/or create an informative and searchable inventory of all data assets in and out of an organization in connection with the AInabler platform.

Security is vital putting these pieces together. ESA operational environment enforces it explicitly by a separation between the operations LAN (which is segregated with high level security) and the pre-operations LAN (which has a lower level of security segregation. This LAN segregation impose strict controls on the data that can be exchanged between them and with external entities. This division is also being considered in the platform DevSecOps, in addition to the data accessibility and cataloguing.

4.1 AI4OPS External Interfaces

The AInabler Platform exposes to the AI Platform Data Streaming Component, to the AI training module and to external users (either through a dedicated GUI or directly programming through the API); a list of generic functions for communicating with the available MOS through a RESTful web service interface API.

This interface and its implementation have been designed to access all types of Operational Data in an agnostic manner. All data are sent and delivered using JSON format. It is based on a stateless request – response pattern where, in this case, the interface provided by the AI Platform manages all incoming requests and provide suitable responses.

This interface is provided by the Request Manager component which serves as a middleware between the Operational LAN and the Platform, where dedicated endpoints are implemented as different microservices to provide access to different MOS. Each MOS Provider is designed as its own microservice created upon each request (once the HTTP Basic Access Authentication scheme has been performed) and ending after publishing the result into a specific queue which later is forwarded to the original caller (e.g. notebook, AI model, user request).

17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

Currently extension points are provided for some of the existing ESA Mission Operational Systems (i.e. MUST, ARES, SCOS-2000, EGS-CC). However, new ones can be created if new interaction types are needed in the future.

4.2 Result Report Manager REST Interface

Exposure of the data contained in the platform mongo DB related to the results of the AI applications has been deemed needed to be also exposed externally to the platform. In this manner users and applications can benefit from previous AI runs to analyse and, in the future, retrain their models.

The provided interface is based on the REST services , so all data are sent and delivered using JSON format. In line with the rest of the platform architecture, it also follows the request – response pattern, where, in this case, the interface provided by the Mongo DB AI Platform manages all incoming requests and provide suitable responses.

5 Implemented Use Cases

Showing AI can bring real operational benefit and be generally applicable to many, if not all, of ESA's missions; rather than just solving niche problems specific to a single mission; is needed for the adoption of a new technology across the domain.

Therefore, 3 initial use cases applicable to different mission families, objectives, and time schedules have been implemented using the platform with the objective of demonstrating how it can be used to solve actual issues relevant for different operational teams. These cases are described in the following sections.

5.1 Contextual Novelty Detection

This use case describes an extension to the Novelty Detection [13] approach for intelligent monitoring. Unusual behaviour is often the signature of an anomaly on the way to happen. By detecting unusual behaviour early, operators have time to investigate and take preventive actions before the anomaly fully develops. This significantly reduces the risk of the mission and provides peace of mind to the flight control team.

This use case's application enables flight control engineers to become aware of unusual behaviour with the granularity of TM parameters when considering the whole spacecraft status as context. This contribution significantly complements individual parameter anomaly detection approaches. Contextual Novelty Detection is computationally expensive, especially with big amounts of data. For this reason, in a first approximation an interplanetary mission, Mars Express (MEX), was considered due to their reduced sampling rate. Among the current ESOC mission, MEX is the interplanetary mission with the longest duration.

5.1.1 AI/ML Approach for the Contextual Novelty Detection Application

A gradient boosted tree Machine Learning model is trained for each TM parameter to be monitored. The task of each of these TM models is to predict what the values for each TM parameter should be. As input, it takes other TM parameters, Telecommands, events, etc.

The output is compared to real values and significant deviations would signal a non-expected behaviour given the context. The problem is framed as a set of multiple self-supervised tasks (one per TM parameter). The term "self-supervised" is used to indicate that supervised machine learning models are used without any effort to label data.

Fig. 4 below, shows a simplified example of the input data of the Machine Learning model during learning and prediction phase as deployed on the AInabler platform. The blue parameter APR_1_D is the target parameter that is predicted. During the learning phase, all data (TM, TC and EV) including the target parameter is available. In the prediction phase, the target parameter is removed from the input set and predicted based on the remaining data. The resulting values are then compared to the real values of the target parameter to determine possible deviations.

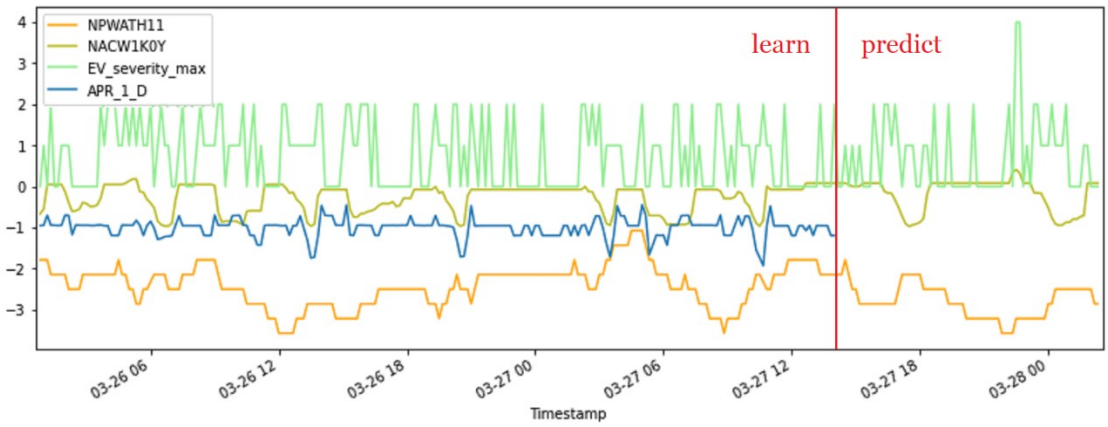

Fig. 4 Input data of the Machine Learning model during learning and prediction phase

To capture all connections (including the obvious ones) every other TM parameter, TC and S/C event are considered as input. Also, non-obvious connections are identified by discarding those input parameters which are highly correlated with the output. Two models can be trained for some TM parameters (including-obvious and non-obvious) and check with users in which circumstances one model is better than another.

When it comes to predict a single TM parameter, it is foreseen that not all the other TM parameters, TCs and Events are needed. However, it is hard to tell in advance which inputs will be necessary due to the non-linear couplings. For this reason, it is proposed to simplify the ML models in a second stage, once that it is known which input each of them needs.

The output of the application consists of the parameters that were found to show a novel behaviour, the start and possibly an end date of the detected novelty, a severity or confidence score as well as the direct and indirect input parameters (TM/TC/EV) that were involved in detecting the novelty. The list of anomalies is forwarded to the AInabler Kafka stream and stored in the platform's MongoDB database. MUSTLink has been extended with capability to retrieve anomalies from the platform through the platform's external results API. WebMUST can then visualize the anomalies either in a table or on a timeline plot as shown in Fig. 5 below.

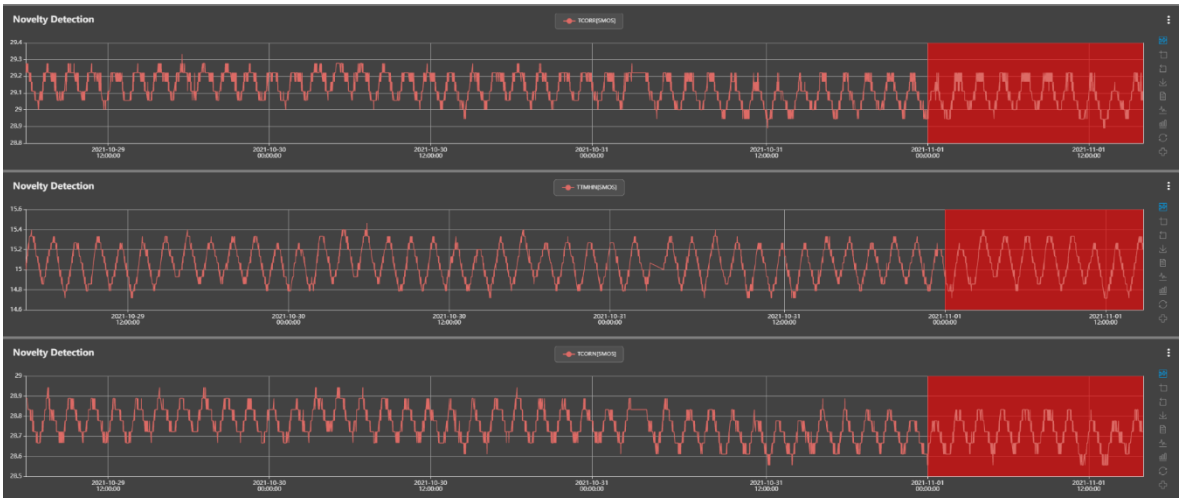

Fig. 5 Novelties displayed on a telemetry plot on the WebMUST dashboard user interface.

5.2 Time Series Prediction application

Time series prediction is a method of predicting future values by observing historical data. A first approach consists in forecasting for a single time value step. This can be done either for a single parameter or for multiple parameters.

It is also common in time series analysis to build models that instead of predicting the next value, predict how the value will change in the next time step. This can be done either by making the predictions all at once (Single-shot) or by making one prediction at a time and feed the output back to the model (Autoregressive).

This use case is focused on supporting operators in near-real time, providing forecasts of the evolution of the mission and supporting decisions making as they understand better the future possible values and their uncertainty.

5.2.1 AI/ML Approach for Time Series Prediction

Different algorithms are available for time series prediction. For instance, the "autoregressive integrated moving average" (ARIMA) approach is among the most widely used approaches for time series forecasting. For this Use Case, the Temporal Fusion Transformer (TFT) [14] technique was selected as the most promising, as it allows for time series prediction at multi-horizon, needed to provide an accurate prediction of not only a point in the future, but also, how the future may evolve. TFT is an attention-based architecture which combines high-performance multi-horizon forecasting with interpretable insights into temporal dynamics. To learn temporal relationships at different scales, TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies. TFT utilizes specialized components to select relevant features and a series of gating layers to suppress unnecessary components, enabling high performance in a wide range of scenarios [15]. In addition, predicting time series with TFT is a good example on how the AInabler platform can be used to train and serve deep learning models.

While this use case can be applied to all ESOC missions, these methods work best with data that contain certain periodicity (plenty of examples can be found here as many TM parameters are heavily involved by the orbit, especially Earth Observation missions). The GAIA, as astronomy mission was selected as representative use case, as it also has periodicity connected to daily operations. The use case is also intended to also our intention to check the forecasting performance on non-periodic sensor readings.

Telemetry data in the form of engineering TM parameters are the most useful for this use case. The ARES archive is used for data retrieval. The output of the application consists of a time series prediction (multiple time values) providing also intervals of confidence of the prediction (Fig. 6). Probabilistic models provide with insights into the uncertainty of a forecast.

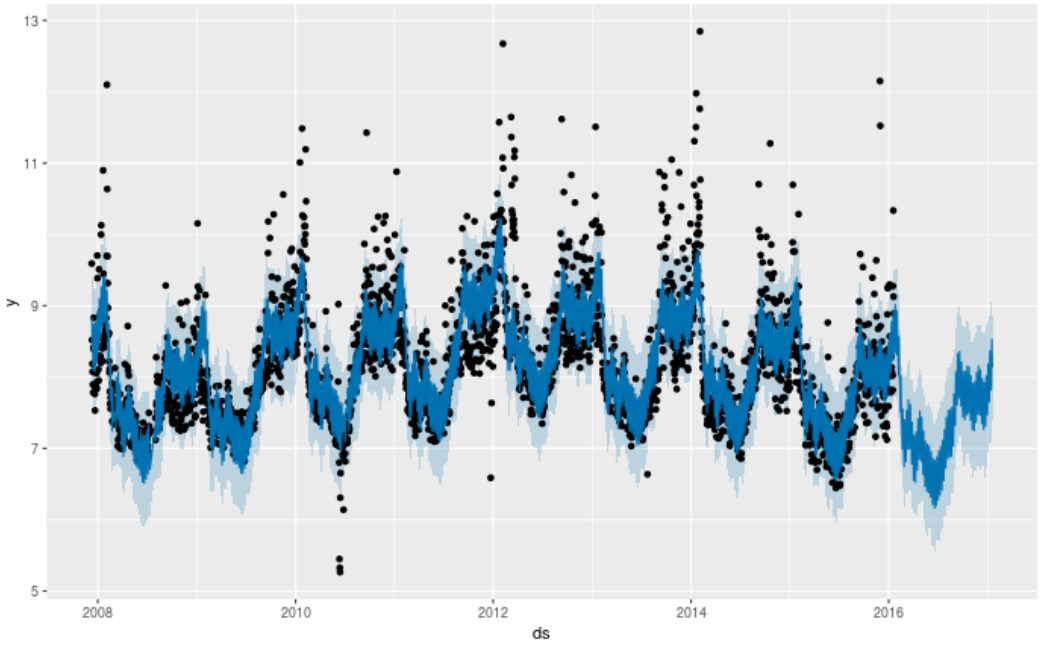

Fig. 6 Time Series Prediction Use Case

The application obtains input data through the ML platform's request manager, which makes the request to MUSTLink using the API Manager. MUSTLink's data aggregation functionality is used to speed up the request by only requesting the average value in a time range, e.g. 10 minutes, instead of retrieving all samples.

The pre-processing stage receives the TM data in MUSTLink's JSON format and transforms it a pandas series. As deep learning techniques need data to have small ranges (to prevent from saturation), the TM data is scaled. Training the model starts with the data that is retrieved and pre-processed. TFT implementation from darts then is used to learn to predict the future evolution of the TM parameter of interest.

The application output data which consists of the predicted timeseries and confidence interval defined by high and low thresholds is forwarded to the Kafka stream and stored in the platform's MongoDB database.

5.3 Automatic Anomaly Reporting Application

This use case can be seeing as an extension to the Novelty Detection approach for intelligent reporting. The current Novelty Detection has the benefit of providing exhaustive and fine-grained information about the status of a spacecraft system at parameter level. However, it comes with the disadvantage of overwhelming operators when anomalies affect many TM parameters. It is often the case that an anomaly can affect the behaviour of many TM parameters. For example, a collision with a micro-meteorite with a solar array can change the behaviour of voltages, gyroscopes, temperatures, GPS signals, etc. Each of these subsystems have many TM parameters that will present a new behaviour in the TM readings. Having a big list of potential anomalies to check is discouraging to flight control engineers as it takes effort to check each of the potential anomalies. In addition, it erodes the trust on the anomaly detection results with the possible consequence of overlooking future relevant detections.

The goal of this use case was to effectively report anomalies to operators. In contrast with the current approach, operators are provided with a higher level of abstraction (with the possibility to investigate the details if needed). This can be seen as an intelligent post-processing of the anomaly detection results.

The missions to benefit the most are those that regularly use Novelty Detection. For this reason, SMOS was selected as target mission. In fact, Novelty Detection results of this mission are closely and regularly monitored, and the nominal periods are kept updated.

Telemetry data is used in the form of engineering TM parameters. In addition, information about previously reported novelties is also used as input. The previously reported anomalies refer to those anomalies detected by the current anomaly detection approach used by ESOC and are retrieved through MUSTLink by using the novelty detection endpoint.

The output data of the application consist of list of anomalies and corresponding metadata such as anomaly start/end time, list of parameter clusters related to each anomaly, and the parameters belonging to each cluster.

5.3.1 AI/ML Approach for Automatic Anomaly Reporting

The approach builds on the results from Novelty Detection systems and is not tied to any particular novelty detection algorithm implementation or approach (other than assuming that novelty detection is performed at parameter level).

In order to report the novelty results in a more effective way, the parameters with similar behaviour are grouped by using unsupervised learning. The following possibilities are considered:

- Correlation analysis: the groups are formed by parameters having a high correlation. This might pose the problem that 2 parameters might not be directly correlated but only though a third parameter (depending on the chosen correlation threshold).
- Time series clustering: the groups are formed by the parameters belonging to the same cluster. There are several clustering algorithms that can be considered. At this time, we can think of clustering approaches that do not take the number of clusters as an hyperparameter to allow for the emergence of a natural number of clusters depending on each of the results.

After assessing the two most promising options (correlation analysis and time series clustering), it was decided that time series clustering would lead to better groupings. The main reason supporting this argument is that correlation analysis can only find linear relationships while clustering can find both linear and non-linear relationships. In addition, by using clustering, unsupervised learning capabilities within the platform can be demonstrated.

From the available clustering algorithms available, the “Ordering points to identify the clustering structure” (OPTICS) algorithm [16] is the most scalable one allowing for a very large number of samples and the possibility of having large clusters. In addition, it features very desirable properties applicable to our use cases such as dealing with uneven cluster sizes and handling variable cluster density. OPTICS is transductive machine learning method, meaning that it is designed to model a specific dataset, but not to apply that model to unseen data. In this use case this is not a problem as there is no need to assign to an existing group a new TM parameter.

The ML approach consists of finding clusters in TM parameters (time series) using the OPTICS clustering algorithm. These clusters will then be used to group the anomaly detection results, making it easier for operators to interpret the results while reducing the information overload. For example, an anomaly affecting any of the parameters in the next figure will likely affect all the others in the same cluster. By grouping the anomaly in a single behaviour, operators receive less alarms and they reduce the need of investigating anomalous behaviours. They can, of course, get the full list of detected novelties if interested.

The application output data which consists of a list of anomalies and their parameter clusters.

6 Community collaboration

The AInabler software source code, documentation, example use cases and development environment are made available through the Space CODEV platform [17] within the territory of the ESA Member States and under a permissive software license allowing commercialisation of the software. Space CODEV is a collaboration platform of the European Space community born from an ESA initiative. The collaborative software development platform is

based on the Gitlab platform and provides means to contribute to the software used by a community of partners including ESA.

7 Summary

The AInabler platform developed in the frame of the ESA GSTP Study *"Ground Segment Operations Automation Using Artificial Intelligence (AI4OPS)"* has been established as a common environment for developing and supporting AI-based applications for operational use cases. It enables its users to build, train, and deploy ML models of AI applications to support mission operations and provides easy interface to data from ongoing or past space missions.

The platform software and documentation are made available through as ESA community license. The goal is the provision of a "platform as a service" (PaaS) available to internal ESOC projects and to data and software developments driven by ESA affiliates, contractors or other 3rd party collaborators and serving as a key building block in ESOC's AI Roadmap for increasing the automation in mission operations.

The platform design follows the MLOps chain, automating much of the lifecycle of AI applications and ML models, allowing the deployment of the applications, the continuous integration (CI), continuous delivery (CD), and continuous training (CT). Within the AI4OPS project the platform was implemented up to level 1, focusing on the initial automated CI/CD system, with the evolution to Level 2 being implemented in follow up activities.

The platform is at its core a data platform, including all standard development languages, tools, and libraries, which enables its users to build, (re-)train, and deploy machine learning (ML) models to implement any number of use cases and supports fast prototyping and testing, allowing fast judgment of the feasibility and possible value of a potential model, likewise, inspection and evaluation of data quality.

Once the platform was established, three use cases were identified and implemented as application to test and demonstrate the capabilities of the AInabler platform. The included application for Contextual Novelty Detection, Time Series Prediction and Automatic Anomaly Reporting.

In addition to the platform, as series of documents was established to support the introduction of AI/ML into mission operations teams at ESOC, providing background in relevant topics such as strategies for deploying Artificial Intelligence applications, AI information architectures, operational data acquisition and data access governance guidelines.

The AInabler platform has been successfully deployed in the ESA IT Cloud and will serve as the baseline for the further development and deployment of operational use cases as part of ESOC's AI Roadmap, which intends to use AI and ML to support a high level of automation in its mission operations.

8 References

[1] ESA/ESTEC, "Artificial Intelligence in ESA, v1.3," ESA-TEC-RP-013019.

[2] ESA/ESOC, "AI Strategy," AIS v1.3, 2022.

[3] ESA/ESOC, "AI Information Architecture," ODAcqG_v1.0, 2022.

[4] ESA/ESOC, "Operational Data Acquisition Guidelines," AI4OPS, ODAcqG_v1.0, 2022.

[5] ESA/ESOC, "Operational Data Access Governance," AI4OPS, ODAGov_v1.3, 2022.

[6] European Space Operations Center (ESA/ESOC), "A2I Roadmap - High potential of AI for Space Missions," https://esoc.esa.int/a2i-roadmap.

[7] Google Cloud Architecture Center, "MLOPS Continuous Delivery and Automation Pipelines in Machine Learning," https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning.

[8] Wikipedia, "Lambda Archiitecture," [Online]. Available: https://en.wikipedia.org/wiki/Lambda_architecture. [Accessed Jan 2023].

[9] Google, "ML Metadata (MLMD) Library," https://www.tensorflow.org/tfx/guide/mlmd.

[10] Amazon, "S3 API Reference," [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/API/Type_API_Reference.html. [Accessed Jan 2023].

[11] T. K. Authors, "Kserve Documentation Website," https://kserve.github.io/website/0.9/.

[12] "Strimzi," [Online]. Available: https://strimzi.io/.

[13] ESA/ESOC, "Advanced Proof of Concept AI Application Documentation – Contextual Novelty Detection AI4OPS_APoCAD_CND, v0.1," 2022.

[14] S. O. A. N. L. T. P. Bryan Lim, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," in *arXiv:1912.09363*, 2019.

[15] S. O. A. N. L. T. P. Bryan Lima, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," International Journal of Forecasting, 27 Sep 2020.

[16] M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander., "OPTICS: Ordering Points To Identify the Clustering Structure.," in *ACM SIGMOD international Conference on Management of Data*, 1999.

[17] Space CODEV, [Online]. Available: https://ainabler.space-codev.org/.

[18] "Novelty Detection: A New Telemetry Monitoring Paradigm," ESA/PAT/572.

[19] ESA/ESOC, "Advanced Proof of Concept AI Application Documentation – Time Series Predition AI4OPS_APoCAD_TSP, v0.1".

[20] ESA/ESOC, "Advanced Proof of Concept AI Application Documentation - Anomaly Automatic Reporting AI4OPS_APoCAD_AAR, v0.1," 2022.