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

SESAM: An Experiment for AI-Based On-Board Satellite Monitoring

Mathias Zaroubian[a]*, Pauline Delande[b], Pierre-Baptiste Lambert[c], Lucas Coratger [d], Cecilia Ouriachi [e], Pierre Lefevre [f]

[a] Centre national d’Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France, mathias.zaroubian@cnes.fr
[b] Centre national d’Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France, pauline.delande@cnes.fr
[c] Centre national d’Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France, pierre-baptiste.lambert@cnes.fr
[d] Centre national d’Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France, lucas.coratger@orange.fr
[d] Centre national d’Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France, cecilia.ouriachi@2022.icam.fr
[f] Centre national d’Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France, pierre.lefevre@isae-supaero.fr
* Corresponding Author

Abstract

Telemetry (TM) monitoring is a major issue to maintain satellites in optimal conditions to perform their missions. Various techniques, either on-board or on the ground, are traditionally used to accomplish this goal, ranging from simple Out-Of-Limits (OOL) checks to more complex analyses (in frequency domain for instance) as well as long-term ageing monitoring (evolution of daily telemetry parameter statistics). These techniques aim at either anticipating possible in-flight anomalies (ground monitoring) or preventing failure propagation with the on-board Failure Detection Isolation and Recovery (FDIR) function. However, since it is impossible to forecast during the design phase every single anomaly that could be encountered in-flight, these traditional monitoring methods have shown some limitations, particularly in case of new atypical satellite behaviours. In this respect, Machine Learning (ML)-based approaches are well suited for detection of unknown signatures and can be used to fill this gap. Since 2016, in addition to legacy ground monitoring techniques, a One-Class Support Vector Machine (OC-SVM) algorithm called NOSTRADAMUS has been used to monitor the telemetry of various CNES space missions, such as Earth Observation satellites or the SEIS instrument of Mars mission Insight. In order to continue the work on complementing the existing monitoring methods, CNES is developing SESAM (Space Experiment for Satellite Artificial intelligence Monitoring) to demonstrate the interest and the operational feasibility of an FDIR enhanced by AI-based telemetry analysis. A light version of SESAM, derived from NOSTRADAMUS, is currently being tested on OPS-SAT (ESA 3U CubeSat) as a first proof-of-concept (both in terms of hardware requirements and operability). However, due to platform issues, the application has never run successfully entirely on-board but only on flat-sat achieving detections of 23 telemetry parameters in about 10 seconds. Based on the lessons-learned from this first try-out, a new version of SESAM is being developed aiming at having a more direct interaction with the satellite FDIR and dealing with false detections by comparing NOSTRADAMUS outputs with another ML algorithm. For this reason, a more complete version of SESAM will be implemented directly within the flight software of AeroSat (CNES 3U CubeSat), which will be launched in 2024. This new trial will allow the experiment to engage actions, such as oversampling of telemetry acquisitions or equipment reconfigurations, when an anomaly is detected. The on-board computer of AeroSat executes this new version of SESAM in 35 seconds for 47 parameters. These first experimentations shows the usefulness of a FDIR enhanced by an AI-based monitoring method.

Keywords: Artificial Intelligence, Machine Learning, On-Board Anomaly Detection, Satellite Telemetry Monitoring, FDIR

Acronyms/Abbreviations

AI	Artificial Intelligence
AMS	Anomaly Management System
CNES	National Centre for Space Studies - Centre National d’Etudes Spatiales
CPU	Central Possessing Unit

DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ESOC	European Space Operations Centre
FAR	False Alarm Rate
FDIR	Failure Detection Isolation and Recovery
HKTM	Housekeeping Telemetry
ML	Machine Learning
NOSTRADAMUS	New Operational SofTwaRe for Automatic Detection of Anomalies based on Machine-learning and Unsupervised feature Selection
OBC	On Board Computer
OC-SVM	One-Class Support Vector Machine
OOL	Out-Of-Limits
PUS	Packet Utilisation Standard
RTOS	Real-Time Operating System
SEPP	Satellite Experimental Processing Platform
SESAM	Space Experiment for Satellite Artificial intelligence Monitoring
SMOS	Soil Moisture and Ocean Salinity
TC	Telecommands
TM	Telemetry

1. Introduction

Spacecraft health monitoring is a key element of space operations. From simplest Out-Of-Limits (OOL) checks made on-board in real time by the Failure Detection Isolation and Recovery (FDIR) function to Spectral Power Density analysis performed on-ground a few days or weeks after the telemetry generation, anomaly detection is based on the knowledge of the nominal functioning of an equipment and a good anticipation of its failure modes. Therefore, the common denominator of these rule-based alarms is that each one is designed to detect a specific anomaly signature. Thus, this type of monitoring is not fit to identify atypical behaviours of a telemetry parameter that stays within its surveillance range.

At the same time, the main objective of Spacecraft engineers is to maintain their satellite in operational conditions in order to maximize the mission duration. Consequently, more and more monitoring methods are imagined to reduce mission unavailability and it is in this context that, in the last decade, several approaches based on Artificial Intelligence (AI) have been proposed in the literature with the objective to complement legacy monitoring processes. After a trade-off between several Machine Learning (ML) techniques, CNES has adapted a One-Class Support Vector Machine (OC-SVM) algorithm in order to fill that need [1, 2]. Since 2016, this application, called NOSTRADAMUS (New Operational SofTwaRe for Automatic Detection of Anomalies based on Machine-learning and Unsupervised feature Selection), is used on-ground to analyse the downlinked telemetry and has demonstrated its capability to detect atypical behaviour of CNES satellites.

With the emergence of New Space, the computing power of on-board calculators is growing fast which makes it possible to imagine on-board applications of such algorithms. For this reason, CNES is working [3] on an on-board version of its AI-based health monitoring method: SESAM (Space Experiment for Satellite Artificial intelligence Monitoring) [3]. The main objective of this experimentation is to demonstrate the technical and operational feasibility of on-board fault detection based on AI in order to reduce the latency between the first signs of an anomaly and the first performed actions (compared to on-ground AI-based analysis). Thanks to an interface with the FDIR, the outputs of SESAM can be used to automatically execute on-board actions from oversampling acquisition of a parameter with atypical values, giving more observability to Spacecraft engineers, to equipment reconfiguration in case of a severe anomaly.

In Section 2, this paper describes briefly the utilisation of NOSTRADAMUS in CNES operation centres with a focus on the feedbacks that can be of interest for the development of the SESAM experiment. Section 3 then presents the first version of SESAM that has been developed to be run on OPS-SAT. Finally, Section 4 presents the current on-going developments for the AeroSat mission.

2. NOSTRADAMUS

Since 2016, NOSTRADAMUS is used within CNES operation centres to perform on-ground anomaly detection on satellite telemetry. More precisely, it aims at detecting the first signals of an abnormal behaviour that would not be detected by other more "classic" on-ground monitoring means and that may ultimately result in equipment or system failure.

2.1 Functioning presentation

This ML application relies on a mathematical classification method: an unsupervised OC-SVM algorithm [1]. This approach consists in classifying the analysed telemetry into two categories:

- The normal class if the parameter is close enough to previously encountered data (assumed to be nominal),
- The abnormal one if the sample is too different (which indicates a new atypical behaviour).

To do so, in **learning mode**, a training dataset allows the production of models which consist of a decision frontier that can be used by NOSTRADAMUS to estimate, in **detection mode,** if the new telemetry contains abnormal behaviours [3].

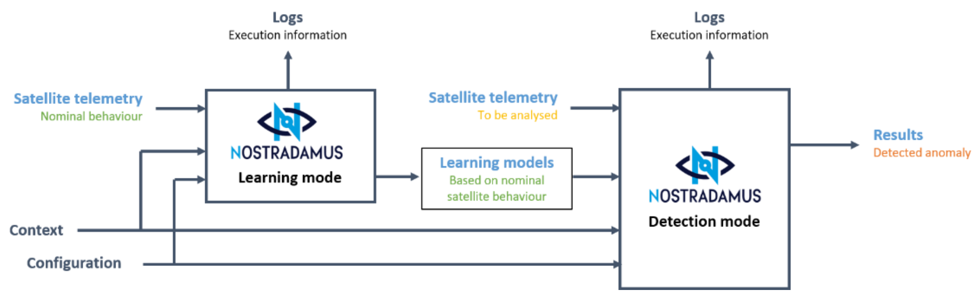

FIG. 1. NOSTRADAMUS high-level operation diagram

After a segmentation into time-windows of the dataset, NOSTRADAMUS associates a 12-dimensional vector (composed of the mean, maximum, minimum, standard deviation, skewness, kurtosis, energy and frequencies features) to each window. In learning mode, some of these vectors (the reference ones) are chosen to "support" a non-linear frontier that becomes the decision frontier of the model. In detection mode, NOSTRADAMUS detects an anomaly (atypical behaviour) when the newly tested vector is outside the decision frontier. Then, the distance to this boundary is normalised between 0 and 100, which represents the anomaly score.

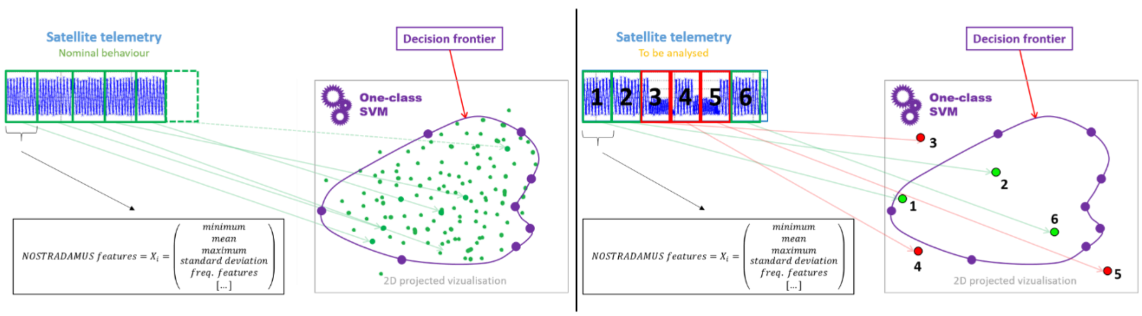

FIG. 2. NOSTRADAMUS Learning and Detection mode representations

2.2 Operational utilisation

NOSTRADAMUS is being used on several missions operated by CNES, including Earth Observation satellites since 2016 and more recently scientific Spacecraft and Mars instruments. The objective of this algorithm is not to replace but to complement the already existing on-ground monitoring methods to detect new types of anomalies or abnormal behaviours.

In order to use this ML application, several operational constraints must be discussed. First, to reduce the total number of false alarms, and so to limit their analysis, the baseline for Spacecraft engineers is to select about **100 relevant telemetry parameters** per satellite for examination by NOSTRADAMUS. After, as mentioned, the dataset is segmented into time windows. Their size highly depends on the mission but a typical configuration is to use **24 hours time windows** for telemetry sampled at 1/32Hz. The training mode is performed every **3 or 4 months** (i.e. before the data begins to deviate too much from the models) on about **12 months** of nominal telemetry. A threshold on the anomaly score is defined to filter detections of NOSTRADAMUS in order to optimize its precision. A value of **40** gives good results [3]. The detection is performed **once a week** on the selected parameters of the telemetry downlinked the week before.

The main objective of this paper is not to precisely describe the tuning of NOSTRADAMUS. All discussions about these values can be found in past papers about this algorithm [1–3].

2.3 Interest for embedded application

ML algorithms are a relevant approach to complement classical Spacecraft monitoring methods for the detection of unanticipated comportment of an equipment [1, 2]. However, one of the major limitation of the actual utilisation of NOSTRADAMUS is the delay between the atypical behaviour occurrence and its detection by the AI tool. In fact, the analysis of the telemetry is performed in deferred time, after the TM downlink, up to a complete week after the actual operational occurrence. In the best-case scenario, this delay can be reduced to a few hours (depending on the frequency of the telemetry download). Therefore, in order to anticipate severe anomalies and limit their impacts on mission availability, this interval shall be scaled down even more.

The objective of monitoring is to react as quickly as possible in case of a failure. Therefore, the telemetry analysis being realised on-board, there is an interest in performing an action in case of an anomaly detection. To do so, SESAM must be interfaced with the FDIR system that is capable to trigger actions. Examples of actions include TM oversampling to improve observability, or even equipment or satellite reconfiguration.

FIG. *3* presents an atypical behaviour remained undetected by classical monitoring methods. In that particular case, an implementation error in the thermal regulation algorithm induced a reconfiguration in safe mode of a satellite a few months after this first behaviour. An on-board AI-monitoring such as SESAM could have detected this first premises and performed an oversampling of that temperature (and on the parameter of the activation of the heater) that would have permitted a better analysis by Spacecraft engineers, an understanding of the anomaly and the avoidance of the reconfiguration that suspend the mission during several days.

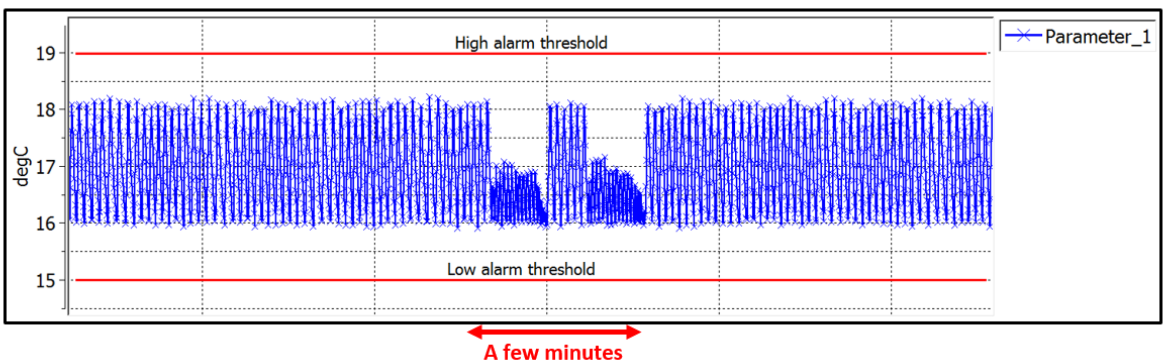

FIG. 3. Example of undetected anomaly on the thermal regulation

For this reason, SESAM is actually under-development by CNES with the objective to demonstrate the interest and the operational feasibility of a FDIR system enhanced by AI-based telemetry analysis. A light version of the SESAM experiment is currently being tested on-board OpsSat (ESA) and a new version will be implemented on AeroSat (CNES) to be launched in 2024.

2.4 Interesting feedbacks for embedded application

The feedbacks of the 6+ years of on-ground operational utilisation of NOSTRADAMUS brings several points of attention for the development of SESAM.

Firstly, the duration of an analysis by NOSTRADAMUS of one hundred parameters is not negligible even on basic office laptop. At the same time, in real time softwares, the execution duration of an application must be precisely characterised and restrained. Therefore, the exact execution period of SESAM must be determined and reduced as much as possible. It is obvious that this value will highly depend on the number of analysed parameters, which could be considered as a variable to tune the execution duration if needed. Indeed, the final objective is to complement the FDIR that must have a really quick response time: the execution of SESAM must be as fast as possible (even on an on-board processor less powerful than the ones available on-ground).

Also, classification algorithms such as the OC-SVM relies on the repetitiveness of the telemetry. In fact, the main objective is to detect new atypical behaviours, which implies that the data must be composed of recurring patterns. These repetitive shapes define a consistent model that allows the observation of slight modifications of the signal. This means that missions with non-recurring telemetry data are not suitable for this kind of ML applications. For example, if the activation frequency of a payload varies a lot during the Spacecraft life (it was the case for the Mars seismometer SEIS), the accuracy of NOSTRADAMUS monitoring will decrease. This is also a constraint for some LEO missions, particularly for dawn/dusk orbits, because of the variation of the eclipse duration along the year. In some cases, one could advise to select one model for the period with eclipse and one for the rest of the year as it is done on several missions like SMOS (Soil Moisture and Ocean Salinity). Therefore, the design phase of SESAM embedded application must consider this possibility as an operational constraint.

Linked to this, a major operational issue lies in the frequency of the learnings. In fact, as explained, operationally, to take into account ageing and seasonal effects, learning is performed every 3 or 4 months. The on-board upload of the new models should be optimised in order to update it regularly.

At the same time, for every learning, a lot of nominal telemetry is needed (about 12 months is currently used on-ground). This is not compatible with on-board learnings because the stored data is limited on a Spacecraft. In addition, in early phases of the mission, the lack of telemetry should be an issue. To solve this problem, ongoing works on Transfer Learning [4] are in progress.

Then, two parameters are predominant on the precision of the ML algorithm: the time-windows size and the detection threshold. After a tuning on a test dataset based on satellites currently monitored by NOSTRADAMUS, the best results were obtained with a 24h time-windows and a threshold of 40 [3]. These two parameters must be reconsidered during the SESAM design phase to reach the best possible tuning for each mission.

However, with the current tuning of NOSTRADAMUS, up to 3% of false detections can be raised. For an on-board FDIR application, this is not acceptable. In fact, incorrect actions such as reconfiguration cannot be performed as it could interrupt the mission. Methods to mitigate this issue during the development of SESAM are presented in section 4.2.

Lessons learned during the 6+ years of operational use of NOSTRADAMUS allows considering an embedded version of the OC-SVM-based tool but with some adaptations of its settings to suit to the new use case. The conclusions of this work are detailed in the section 4.

3. SESAM proof of concept on OPS-SAT

OPS-SAT, the first CubeSat designed by ESOC, is a safe experimental platform that flies in a LEO dawn-dusk orbit. Injected into orbit on 18th December 2019, this Spacecraft provides as a Payload a low cost in-orbit laboratory available for authorized experimenters to test, demonstrate and validate their software experiments. Furthermore, the satellite experimental computer running these various experimentations is ten times more powerful than any current ESA Spacecraft. In that context, the idea to perform NOSTRADAMUS detections on-board arose.

After the presentation of the SESAM application, its architecture and the results achieved on flat-sat and Spacecraft are detailed.

3.1 SESAM presentation

With the first positive on-ground results obtained by the operational utilisation of NOSTRADAMUS and the growing interest for embedded machine learning applications such as telemetry monitoring, the possibility given by ESOC to develop and embark a software experiment on a satellite gave birth to the idea of AI-based on-board anomaly detection.

The final objective for CNES is to enhance the legacy monitoring methods used by the FDIR. However, the first step is to prove the feasibility of such an application. To do so, the opportunity given by OPS-SAT allowed the beginning of the development of the first version of SESAM: an embedded version of NOSTRADAMUS. After a porting of NOSTRADAMUS code from Python to Java, an implementation of the detection part of the code to the ESOC satellite environment has been performed. The baseline is to generate models with an on-ground version of NOSTRADAMUS, analyse the telemetry directly on the payload computer of the Spacecraft and log the anomaly detections on files downlinked to the ground for analysis. More precisely, only the detection mode is performed within the satellite (the learnings are realised on-ground by the Spacecraft engineers). No reconfiguration action is triggered on-board as the satellite segregate the experiment code execution and the platform in order to prevent fault propagation.

3.2 Application architecture

3.2.1 OPS-SAT presentation

OPS-SAT is the world first mission dedicated to testing satellite control technology in orbit, aiming at demonstrating improved mission control capabilities. This 3U CubeSat developed by the European Space Agency (ESA) contains an ARM dual-core Cortex-A9 MP Core as experimental computer (800MHz, 1Gb, SD Card 16Gb) which allows to obtain a computing power uncommon in flight. Combined with a large range of payload equipment (cameras, gyros, accelerometers, magnetometers, reaction wheels, magnetorquers, star trackers…), OPS-SAT is a flying “laboratory” that makes it possible to test new control systems and softwares, including AI experiments.

FIG. 4 shows the two main subsystems of OPS-SAT: the platform composed of the main on-board computer and the payload made up of AOCS equipments, communications systems, an imager and the CPU available to demonstrate software experiments (SEPP: Satellite Experimental Processing Platform).

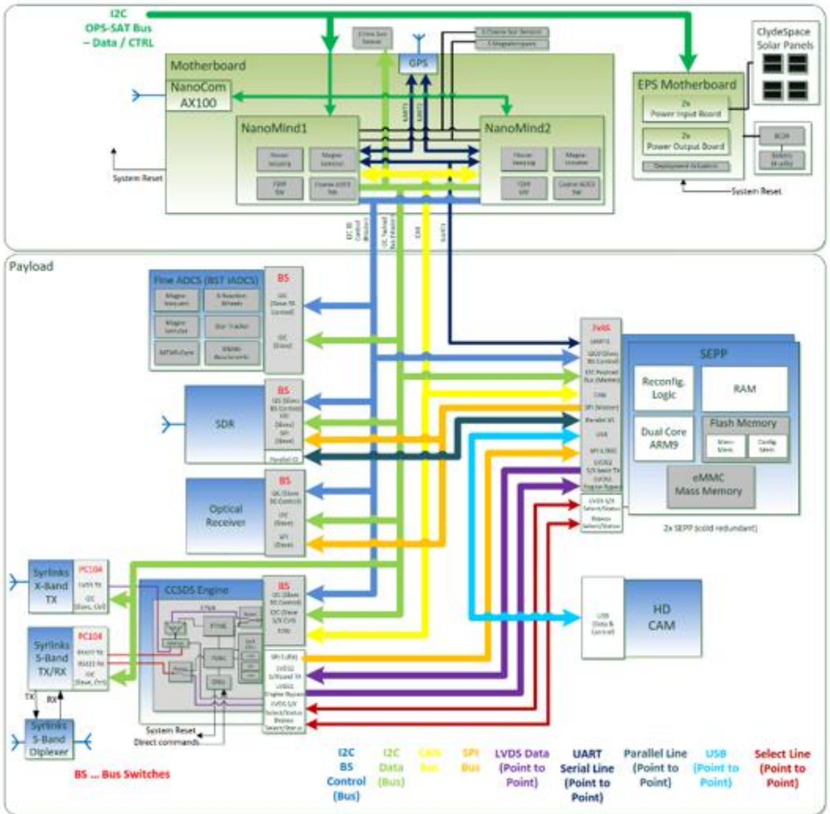

FIG. 4. OPS-SAT architecture

The satellite creates a framework in order to allow the running of application on the SEPP based on CCSDS Mission Operations services. This environment aims to facilitate not only the monitoring and control of the nanosatellite software applications, but also the interaction with the nanosatellite platform. No specific operating system is required for defining this Mission Operation Framework. The minimum system requirement is to have a Java Runtime Environment capable of running Java Virtual Machines. This functioning takes inspiration from the popular concept of apps existent in today's mobile devices. It includes a supervisor application to start/stop applications, to provide a mechanism for finding and connecting to them, and exposes high-level services to monitor and control the nanosatellite platform. In addition to these high-level services, each application may require the development of several extra services.

3.2.2 SESAM Functioning

The first version of the SESAM application has been developed in Java to operate NOSTRADAMUS directly on-board OPS-SAT with the aim of demonstrating the capability to run an AI algorithm dedicated to telemetry monitoring directly on-board a Spacecraft. Even if the architecture of OPS-SAT did not allow it directly (the platform and the CPU running experiments were voluntary segregated to protect the integrity of the satellite in case of dysfunction), the other objective of this experiment was also to open the way for a future complete AI-powered FDIR.

FIG. 5 presents the interfaces between SESAM (or in that particular case NOSTRADAMUS) and the satellite. The TM is made accessible to the application through a telemetry database in which several parameters such as AOCS equipment temperatures, power consumptions and modes are aggregated. This database can be downloaded to the ground in order to perform learnings.

In addition to the upload of these models, the application as well as all required libraries must be uploaded during OPS-SAT passes. After a compression, the SESAM code and additional files compose a package of 3.14Mo that needed three passes to be uploaded on-board (only 1Mo per pass can be transfered by experimenters).

In order to be able to realise detections, several services have been developed. Firstly, an archive service permits queries, retrievals and deletes of objects (such as variables) stored in the COM database. The main interest is that objects are not deleted if the application is stopped unlike Java variables. Thus, if an error occurs on board of the satellite and applications need to be restarted, operations can resume where they left without needing to be reset. After, the files exchanged between the ground station and OPS-SAT are compressed because of the data rate. A service has been developed to realise compression and decompression directly on-board. To finish, a schedule service allows to perform time tagged actions, enabling to elaborate a mission planning that launch runs of SESAM detection sequentially.

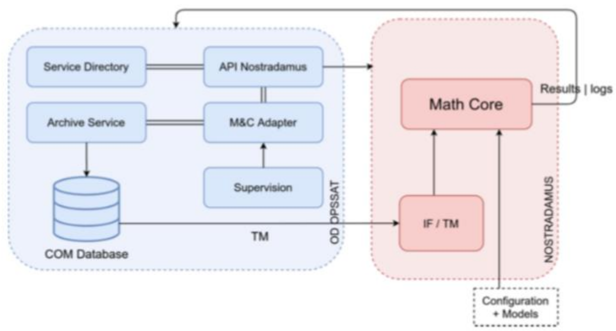

FIG. 5. Logical scheme of SESAM application on-board OPS-SAT

3.3 Operations procedure

On OPS-SAT, each experiment has a time slot (several days) for its demonstration. So, SESAM experiment must be uploaded before each run. When the application is on-board, a run of SESAM is scheduled between passes with the objective to upload a mission plan that allows downloading telemetry (in order to performed learnings), launching the detections and preparing the results log.

On board, after the recuperation of the TAI date from the GNSS receiver, the configurations files are unzipped to prepare the algorithm. A first part of the code permits to retrieve the telemetry from the different equipment to store it within the COM database. The first phase of the experimentation is composed of a creation of a telemetry database to perform learnings. In fact, the OPS-SAT architecture provokes several differences between the housekeeping telemetry downloaded by the Spacecraft platform (and generated by the main on-board computer) and the one managed by the SEPP unit. For this particular reason, learning mode can not be realized on the housekeeping telemetry accessible on-ground and the COM database must be downloaded to generate models.

After the upload of these models, the detection phase composed of a retrieve of telemetry from the COM database and a detection can be launched sequentially until the next pass. Several minutes before that pass, the log files and detections results are compressed to be sent to the ground.

These steps are realized after each pass dedicated to the upload of a new schedule.

3.4 Lessons learned

The first version of SESAM on OPS-SAT reveals several difficulties that are interesting to consider for the implementation of a new version on AeroSat.

To begin, the fact that the housekeeping telemetry was not comparable to the one accessible on-board by the application made difficult the realisation of correct models because of the lack of data. In fact, only the data aggregated within the COM database during the period of the experiment (several days because of number of experiments) can be used as learning dataset. This fact has an impact on the downlink data rate because provokes the transfer of the same information (for instance an equipment temperature) twice to the ground. In addition, in comparison with the 365 days required by NOSTRADAMUS on-ground, this limited dataset deoptimizes the precision of the application. This limitation imposed a reflexion on the access to the telemetry on AeroSat with the choice to use a dedicated buffer in which the telemetry is retrieved from the main HKTM (Housekeeping Telemetry) buffer.

After, the amount of operational actions required from Spacecraft engineers to run this experiment on OPS-SAT is too large to envisage a real operational utilisation of an AI-based monitoring. In fact, the limited duration of the experiment (several days) permits to realise many operations (downlink of the data, learnings, upload of the models and schedule, downlink of the logs…) to monitor the execution of SESAM. However, in operations, all these steps must be automated to reduce the impact on operators. One of the feedback is so to reduce the procedures to only the upload of new models or the modifications of some parameters, and only when genuinely justified. The idea is not to have to upload a schedule every days as it is required on OPS-SAT.

3.5 Results

This version of SESAM has been tested on flat-sat, in collaboration with the ESOC team in charge of operating OPS-SAT. These tests demonstrated an execution time of about 10 seconds on 23 parameters (60 time-windows of 3 minutes, with a sampling period of 3s). Most of this duration corresponds to the execution of the NOSTRADAMUS core. All other steps (compression, aggregation of the telemetry…) remain short in comparison.

The inflight tests of this version of the SESAM experiment are now in progress. Several attempts have been performed but satellite anomalies have limited, for now, the results. Nevertheless, OPS-SAT has already made it possible for CNES to work on best design practices for such applications as well as for operational concepts associated with on-board AI monitoring algorithms. All these results have already proven valuable for future versions of SESAM which are currently in development for AeroSat.

4. SESAM on AeroSat

For a decade, CNES "Nanolab Academy" program has been trying to create an innovative environment to teach students the different Space professions. Its objective is to frame future engineers into the development and operations phases of nanosatellites either within universities or within CNES Space Centre.

After the success of EyeSat (a 3U CubeSat) launched in 2019, the first Spacecraft developed entirely in CNES by over 250 interns and work-studies trainees, Nanolab Academy has begun the development of its next mission: AeroSat. This 3U CubeSat has the objective to perform in-orbit demonstration of six payloads including a new version of SESAM.

FIG. 6 : AeroSat

The development of AeroSat relies on lessons learned during the design and ongoing operations of EyeSat. For this reason, most of the platform remains unchanged. Only the imager and the X-Band transceiver have been removed to allow the new payloads to be embarked.

The actual baseline for the launch is 2024. After the first tries on OPS-SAT, AeroSat represents a great opportunity and interesting perspectives for a new on-board test of SESAM. In fact, unlike OPS-SAT, AeroSat is not designed around an isolated and dedicated computer for experiments. Its architecture is composed of a platform designed to be controlled by a central calculator and of several demonstration payloads. However, NOSTRADAMUS (and SESAM) is a software application and can directly be executed by this central processing unit which allows more possibilities such as a connection with the FDIR. In addition, the execution of SESAM as a part of the flight software offers an access to all telemetry parameters (about an hundred for this satellite) compared to the limited few of OPS-SAT payload.

This section presents the development status, the various design choices that have been made and their impacts on the architecture of SESAM.

4.1 AeroSat On-Board Computer

One of the main difficulties when one talks about an embedded AI application concerns the processing power. Indeed, to deal with the radiative and thermal environment of space, manufacturers concentrate their efforts on the resilience of the components to radiation damage. Resulting prices slow down their evolutions and creates a gap between the processing power of the basic Earth laptop and space processors. In addition, even on ground, the execution of such algorithm is very resource consuming. For instance, on CNES business laptop, the execution of a NOSTRADAMUS detection (and its human-machine interface) takes approximately 2 minutes for about a hundred parameters. However, with the emergence of New Space, the gap between on-board equipment and ground hardware is narrowing which allows considering the execution of more complex functions on-board.

One of the main component of EyeSat was the on-board computer (OBC): the Ninano (FIG. *7*) designed by Steel-Electronique in collaboration with CNES. This calculator will also be used on AeroSat considering of the correct in-orbit functioning on the two first CNES CubeSat Missions (EyeSat and Angels) with the Ninano OBC. This microcontroller concentrates an FPGA and a Xilinx Zynq 7030 SoC equipped with a dual-core ARM Cortex A9 processor. This widespread CPU (largely used on smartphones and tablets) has a layout that makes it a low consumption and small size high clocking core. In addition, its architecture is compatible with various operating systems such as Linux.

FIG. 7. Steel-Electronique Ninano Processor Board [5]

Compared to computers embarked on previous missions, with its 800MHz clocking, the NINANO provides a computing power that makes possible to consider more complex calculations on-board such as AI algorithms. In fact, even if such applications are very demanding in time and computing power, the section 4.4 shows the feasibility of on-board AI-based telemetry analysis on a representative development board.

In addition, on EyeSat, only one core of the NINANO was used but a study is underway to estimate if the second core can be switched ON on AeroSat. The objective is to determine if SESAM can run on a dedicated core in order to accelerate its execution.

To conclude, the AeroSat OBC makes feasible the execution of an AI algorithm. However, some difficulties remain to consider complementing the FDIR with an AI.

4.2 False alarm impact mitigation

The objective of SESAM is to complement the existing FDIR function that is not capable to detect some atypical equipment behaviours. On AeroSat, the experiment will be connected to the on-board system that raise actions at the occurrence of an anomaly. The main issue concerns the trigger of reconfigurations on false detections that could interrupt the mission. The False Alarm Rate (FAR) of NOSTRADAMUS is about 3%, which is not compatible with an on-board use. In fact, the FDIR aims to optimise the operational availability of Spacecraft and cannot suspend the payload activities based on an erroneous activation. Several choices have been made to mitigate this issue.

4.2.1 *Operational Impact*

Firstly, a manner to deal with this issue is to take it into account within the operational constraints. In fact, the easiest method to prevent false detection reconfigurations is to start the experiment with a primary phase consisting only of a recording of the results of SESAM. During this step, the experiment will analyse the telemetry and log its detections. Afterwards, Spacecraft engineers will evaluate if the actions that have triggered are the correct ones. If not, several methods can be imagined to improve the detections: upgrade models with more telemetry, adapt the filter or the threshold of the detection or in worst cases stop to monitor a specific parameter not adapted to machine learning analysis. At the end of this period that can last from few weeks to few months and if the results are satisfying, the connection between the outputs of the experiment and the FDIR function will be established.

4.2.2 Second AI algorithm selection

Even after an observation phase, it is obvious that the number of false detections of SESAM will remain oversize. In fact, as explained in section 2.4, the magnitude of the false alarm rate of NOSTRADAMUS is about **3%**. That means that for every on-board activation of NOSTRADAMUS, at least about **3 detections** will be raised if we were to monitor 100 parameters. That result is not acceptable for a real-time monitoring method. To reduce this rate and mitigate the risk of reconfiguration on false detections, the decision to append a second machine learning algorithm to SESAM in parallel of NOSTRADAMUS has been made. To select this AI method and to limit the processing requirement increase, the main prerequisite was to have the same inputs and outputs than the OC-SVM algorithm. The objective is to reuse all the pre-processing required by NOSTRADAMUS (division of the data into time-windows, computation of the 12-dimension vectors…) to limit the increase of execution duration. The only difference with the OC-SVM-based application must be the method to make the decision between a normal functioning and an atypical behaviour. After a comparison of several machine learning methods on a test basis created from EyeSat telemetry (as a reminder, the platforms of EyeSat and AeroSat are very closed), the most encouraging solution is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN).

DBSCAN [6] is a data clustering method that is based on the distance between points to estimate if they are associated to a nominal functioning or an atypical behaviour. In fact, this type of algorithm relies on the creation of clusters of data and detects as anomalies the points that do not belong to anyone. This concept can be applied to the 12-dimensional space used by NOSTRADAMUS.

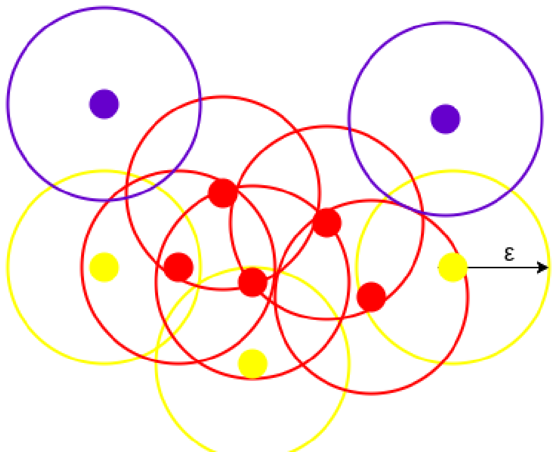

FIG. 8 : Two-dimensions DBSCAN representation. Red dots are the central point and yellow dots are border points of a cluster and the others dots are anomalies (because there is no other point within their ϵ distances).

The main problem encountered with this algorithm is the size of the model files. In fact, a new point is compared with every single point generated from the training dataset. That means that the model contains the 12-dimensions vector of every time-window of the training dataset. The amount of data is much bigger than for NOSTRADAMUS (respectively **36ko** versus **3.4ko** per parameter). The solution is to compress these model files to facilitate the upload on-board. First tries allow a reduction of about 1/3 of the file sizes.

The objective of the addition of a second AI method is to compare the outputs to the NOSTRADAMUS detections, to reduce as much as possible the False Alarm Rate. In order to make them comparable, the outputs of these two algorithms are normalised and an Anomaly Management System (AMS) is required to compute the result of the complete application (section 4.3.3). This value can be used by the FDIR to trigger an action. Several possibilities for action triggering are feasible thanks to this configuration: if only one algorithm detects an anomaly, a simple action, such as an oversampling, can be raised, whereas the detection of both can imply a reconfiguration.

The idea behind the parallelization of two AI methods is to compare the results. It is clear that the addition of more algorithms would reduce even more the false detection rate. However, considering the available processing power, the utilisation of more than two machine learning algorithms is not feasible (section 4.4) but can be consider as a further development. To facilitate the addition of more AI techniques, the architecture of SESAM is designed around this constraint (section 4.3).

4.2.3 Analysis of the severity of the detection

Finally, there remains a lot of work on the specifications for the reconfiguration actions depending on the monitored parameters. For instance, the impact of an anomaly on the payload is not the same as one on the Data Handling Subsystem. If one imagine a problem on the telemetry transceiver, in worst-case scenarios, the Spacecraft can be lost because of an impossibility to communicate with it. For this reason, a reflexion on which actions must trigger on a SESAM detection depending on the parameter in anomaly (or if several ones that are linked are abnormal) must be realised. At the moment, the baseline is to perform subsampling of the parameter (and possibly few others that are directly linked to the one in anomaly) to facilitate Spacecraft engineers analysis and reconfiguration in safe mode for worst detections. In fact, the simplicity of the AeroSat platform does not allow a lot of various actions (not a lot of equipments are redundant for instance) but every possibility will be studied.

4.2.4 Conclusion

In order to reduce the probability of false alarm reconfiguration, several actions are in progress. An operational adaptation consisting of phase without the utilisation of the output of SESAM by the satellite has been decided. In addition, a selection of another algorithm (DBSCAN, in addition to NOSTRADAMUS) was conducted with the objective to realise a comparison of the results of these two methods. To finish, the specification of the event actions (in the sense of the Packet Utilisation Standard) is in progress. The next step is to discuss about the architecture resulting from these design choices.

4.3 SESAM Architecture

Unlike OPS-SAT, AeroSat is partly designed with the objective to demonstrate the feasibility of the on-board execution of SESAM experiment. As explained, the codes of NOSTRADAMUS and DBSCAN will be executed directly on the Spacecraft main on-board computer, the NINANO. This offers more possibilities because of the access to all telemetry parameters and the possibility for a connection to the FDIR. Therefore, various modifications on the application architecture can be identified.

4.3.1 Application Environment

As discussed, the baseline for SESAM on AeroSat is to implement the AI-based monitoring method as a part of the central flight software. In embedded real-time systems, every process must guarantee response within a specific time. This design allows the utilisation of time and space partitioning. Within a given computer, each application software is called a partition and has its own private memory space (other processes cannot write on this zone) and the processing resource in terms of CPU availability or time slot is allocated to it sequentially by an hypervisor (XtratuM [7] in the case of AeroSat). The role of the hypervisor is also to monitor the functioning of the different partitions by killing them if the time slot allocated for them is elapsed.

The objective for SESAM is to develop a Python application running on a Linux distribution contained in a dedicated partition of the on-board software (represented by the blue on the FIG. *11*). Contrary to what has been done on OPS-SAT with the conversion of the NOSTRADAMUS code into Java, the AeroSat's version of SESAM relies on the OC-SVM Python code used in CNES control centres. To run this version, a Python environment (with specific libraries) must be embedded which is made possible by the Linux RTOS (Real-Time Operating System). Another interest of the utilisation of such operating system is that it allows the compression and decompression of data. In fact, as discussed in section 4.2.2, the uplink data rate does not permit to upload the model files without compression. These functionalities must be integrated within the Linux kernel during the development phase.

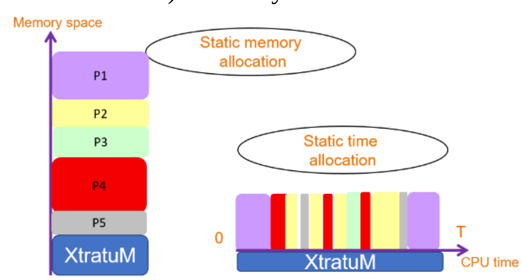

FIG. 9. Time and Space partitioning with XtratuM hypervisor – Each colour represents a partition (or process)

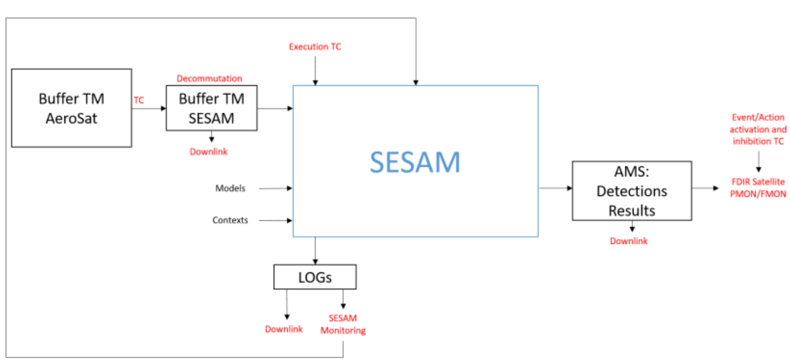

FIG. 10. SESAM Inputs/Outputs scheme (TC: Telecommands, AMS: Anomaly Management System)

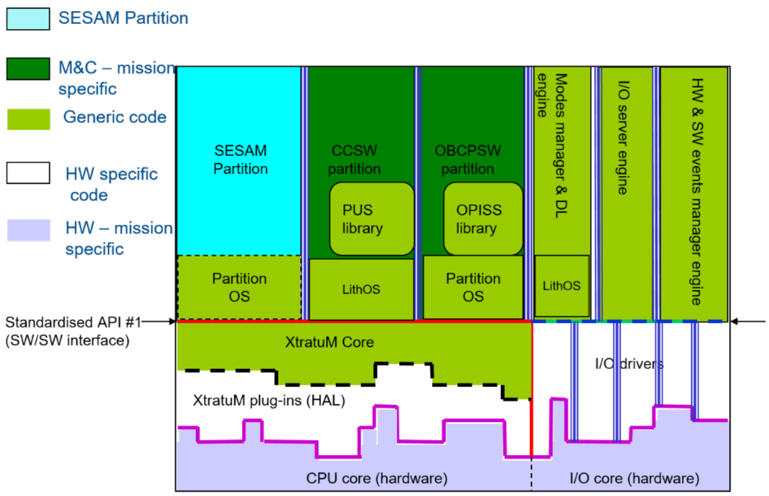

FIG. 11. SESAM flight software environment

For the platform, *FIG. 10* presents the required interfaces of SESAM. In terms of inputs, in addition to the model and context (hyperparameters and some information on the telemetry parameters to analyse) files which are uploaded from ground, only an access to the satellite telemetry through a dedicated telemetry buffer is needed. The application execution logs are saved and downlinked to the ground. After each execution of the SESAM code, the results of both algorithms are compared through the AMS in order to determinate the final detection result that is monitored by the satellite FDIR.

4.3.2 Application architecture

The idea is to make possible the collection of telemetry data from different sensors by SESAM, and then run detections in the dedicated partition using learning models. Like SESAM on OPS-SAT, these learning models will be generated on-ground and then uploaded regularly to AeroSat. The results of the SESAM runs (detection mode) will be stored, retrieved during the following pass and used by the FDIR to trigger various actions.As discussed, SESAM is implemented following the block diagram of the FIG. 12 in Python. The SESAM_core class is the central element of the architecture. It is the link between the two AI algorithms, the different functions that are required to their computations, the AMS and the rest of the on-board software of AeroSat. This main class drives the execution of every detection or learning by demanding access to the telemetry (Get_TM function), and by launching the calculations of DBSCAN and OC-SVM cores, and provides the results of detections to the Spacecraft. It is important to notice that SESAM will have two versions: the on-board one and the on-ground one (presented on FIG. 12). This second one includes the first version in addition to the learning modules.

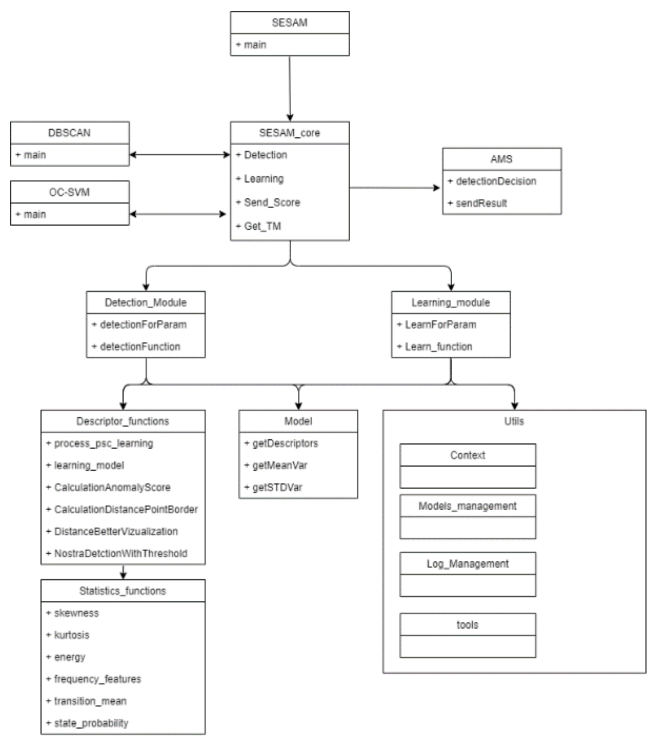

FIG. 12. Block diagram of the SESAM code

It is worth mentioning that the architecture has been imagined around the possibility to use more than two AI cores. The addition of another machine learning method that uses the same inputs and outputs than DBSCAN and OC-SVM is quite simple. In fact, to reduce even more the false detections, the utilisation of at least three different methods is a good compromise (to perform majority vote for instance). However, this possibility has not been studied at the moment because of the limitation of the computing power of the Ninano.

4.3.3 AMS and interface with the FDIR

The outputs of the two algorithms represent normalised anomaly scores. One possibility could have been to let the satellite decide if the parameter is in anomaly considering the two scores. However, the decision to perform this verdict directly in SESAM has been made in order to develop an autonomous software brick that can be easily plug within every on-board software. To do so, the AMS has been implemented with the objective to provide to the FDIR only a global anomaly score.

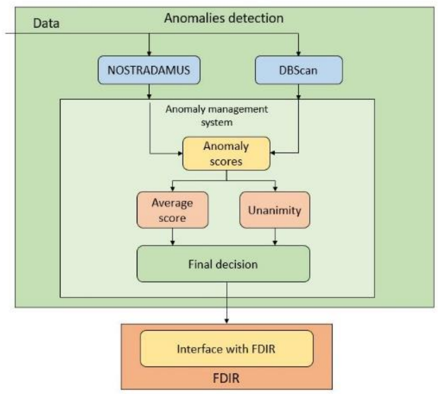

FIG. 13 : Logical scheme of the anomaly management system of the SESAM application on-board AeroSat

Different comparison methods can be imagined. Reconfiguration could be launched if and only if both algorithms (each one being tuned with their own specific threshold) detect an anomaly on the same telemetry parameter during the same time period (unanimity). Another possibility is to calculate the average score of both algorithms and launch a recovery action only if a fixed threshold is exceeded. Now, these two possibilities are feasible with the actual AMS implementation. However, other solutions are feasible and a study on this topic is underway.

4.3.4 *Real-time execution*

SESAM on AeroSat will be tuned with a specific time window but will be launched at a higher frequency (possibly every 5 minutes or so). This particular setting will create a window shift in the detection mode but not in the learning mode. Conducted tests show that this shift has a strong influence on the anomaly score. This challenge is currently being addressed and possible solutions include to optimize the time-window duration or to upload multiple learning models on-board (each one corresponding to a different offset). However, this very last possibility increases a lot the sizes of the model files. A work on the compression methods must also be performed.

4.4 Results in a representative environment

As the mission has not been launched yet, the development of both SESAM and AeroSat is still ongoing. However, first tests have been made in a similar environment, using a ZyboZ7 board [8] that has hardware similarities to the AeroSat on-board computer. This representative framework allows a tuning of the algorithms and a profiling of the Python code.

4.5 Tuning of the parameters

As discussed in the section 2.2, several SESAM parameters has been tuned in order to achieve the best possible results thanks to a dataset created from EyeSat telemetry. Both learnings and detections has been performed on this database. It is important to notice that the hundred parameters of this Spacecraft have a frequency of 1/30Hz.

On models generated from 6 months of data, the most suitable tuning corresponds to time-windows of 12 hours. However, the 600km dust/dawn sun-synchronous orbit of EyeSat and AeroSat imposes the utilisation of at least two models by year (because of the variation of the eclipse duration during a year). This is compatible with an objective of an update of the models every few months.

Now, the anomaly threshold of NOSTRADAMUS (usually set around 40 for other LEO satellites operated by CNES) remains unset because of the health of EyeSat. In fact, the small number of anomalies makes complex the realisation of accuracy tests.

The frequency of the execution of SESAM is highly linked to the results presented in section 0.

4.5.1 *Execution duration*

For a real-time application, the execution duration must be guaranteed. For these reason, a complete profiling of SESAM application has been performed after a code optimization. FIG. *14* shows a linearity of the execution period compared to the number of monitored telemetry parameters. For about half of the total number of AeroSat parameters (47), 35 seconds are required to analyse a 12-hours time-window with the power of a complete CPU core. This execution duration remains longer than the reaction timing requirement for an embedded health monitoring method. However, it should be kept in mind that the objective of SESAM is to complement (and not replace) the existing monitoring approaches that have demonstrated their effectiveness. In addition, the exact number of parameters analysed by SESAM remains to be determined and could be used as an adjustment variable to reduce the execution duration if needed. In addition, SESAM needs 25MiB of RAM to perform this detection.

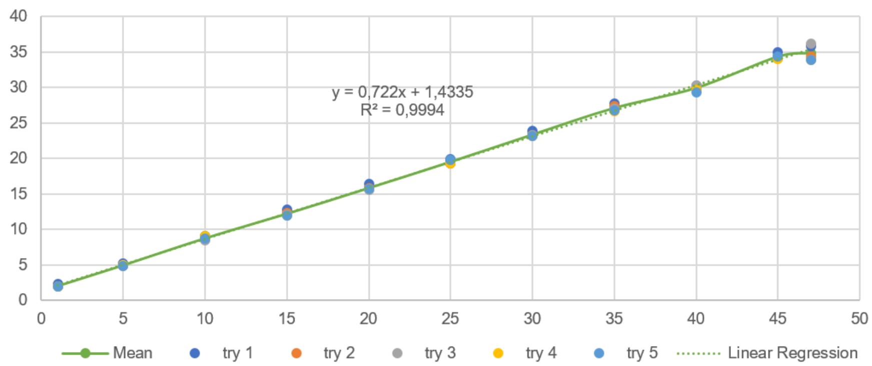

FIG. 14. Execution duration for different numbers of monitored parameters

4.5.2 Further works

The progress of the project shows the feasibility of an embedded AI-based Spacecraft monitoring method to enhance the classical FDIR system. In fact, the AeroSat on-board computer, the NINANO, allows an execution of SESAM in a duration that remains acceptable for a real-time application even if it should be further reduced.

After, there is still a lot of work to be done on the AMS and the different decision-making methods in addition to the determination of the global anomaly threshold. The interface with the PUS (Packet Utilisation Standard) must also be specified.

Then, the requirement of 6 months of telemetry to generate representative models is a strong brake to the utilisation of an on-board machine learning monitoring method. Therefore, CNES is studying the concept of Transfer Learning [4] to solve this issue.

Finally, the profiling performed on the Zybo was executed on a Linux OS directly running on the CPU. To characterize more precisely the application, an implementation within a generic flight software running on the development board will be achieve.

5. Conclusions

The utilisation of CNES OC-SVM algorithm NOSTRADAMUS that analyses in deferred time the telemetry of several satellites showed encouraging results on the interest of ML in operations to complement on-ground monitoring methods. The logical continuation of this work is to implement this algorithm on-board in order to reduce the latency between the new atypical behaviour and its first detection.

The first version of SESAM, a Java edition of NOSTRADAMUS, has been implement in order to be executed on the OPS-SAT Payload computer with a proof-of-concept objective. On flat-sat, 10 seconds were required to analyse 23 parameters. However, because of Spacecraft limitations, no complete execution has been achieved on-board. Several limitations of this first try-out can be observed: the false alarm rate remaining incompatible with real-time monitoring and the segregated architecture preventing reconfiguration actions based on SESAM detections. Nevertheless, the tests on flat-sat remain encouraging for further developments.

After this first trial, the possibility offered by NANOLAB ACADEMY to implement SESAM directly within the central software of AeroSat initiated the development of a new version of SESAM that appends a second ML algorithm, DBSCAN, to the OPS-SAT experiment. Tests carried out on a development board representative of the AeroSat Ninano allow to estimate the requirement in terms of processing power of this new experimentation: 35 seconds of a complete CPU power for 12 hours time-window analysis of 47 TM parameters. Despite the fact that this duration remains too long for the need of responsiveness of the FDIR, SESAM can provide a new analysis point-of-view for on-board satellite monitoring. Nonetheless, the development is ongoing and the implementation within a complete generic on-board software can be a first step to reduce uncertainties of the application.

In parallel of the development of SESAM, an effort is being put into explainability for NOSTRADAMUS, online learning (in order to take into account the feedbacks of Spacecraft engineers into the ML model) and transfer learning (for the purpose of dealing with constellations) [3] in order to improve the functioning of the AI in CNES operation centres.

Acknowledgements

Authors would especially like to thank:

- François Toussaint, Alejandro Diaz, Anthony Surivet, Lucas Coratger, Cécilia Ouriachi and Pierre Lefevre for their work on the development of the SESAM application for OPS-SAT and AeroSat.
- Orphee Faucoz and Felipe Alves Suana for their advices on machine learning methods and their proposition of DBSCAN.
- Julien Galizzi and Simon Corbin for their assistance on the implementation of SESAM within the AeroSat Flight software.
- The Nanolab Academy for the possibility to demonstrate SESAM on AeroSat.
- The European Space Operations Centre (ESOC) from the European Space Agency for the support around the SESAM experiment on OPS-SAT.

References

[1] S. Fuertes, G. Picart, J.-Y. Tourneret, L. Chaari, A. Ferrari, and C. Richard, Improving Spacecraft Health Monitoring with Automatic Anomaly Detection Techniques, AIAA 2016-2430, 14th SpaceOps Conference, AIAA, Daejeon, Korea, 2016, 16 - 20 May.

[2] S. Fuertes, S. D'escrivan, and B. Pilastre, Performance assessment of NOSTRADAMUS & other machine learning-based telemetry monitoring systems on a Spacecraft anomalies database, AIAA 2018-2559, 15th SpaceOps Conference, AIAA, Marseille, France, 2018, 28 May – 1 June.

[3] P. Delande, P.-B. Lambert, M. Bouayad, M. Zaroubian, and E. Jalabert, AI for Satellite Anomaly Detection: On-Ground Operational Feedback and Development of On-Board Experiments, IAC-22-B6.2, 73th International Astronautical Congress, Paris, France, 2022, 18-22 September.

[4] A. Baron, P.-B. Lambert, P. Delande, J.-Y. Tourneret, J. Lesouple, and S. Fabre, Improving AI Monitoring of Early Life Satellites Using Transfer Learning, 17th SpaceOps Conference, AIAA, Dubai, United Arab Emirates, 2023, 6 - 10 March.

[5] Steel-Electronique, NINANO - On Board Computer for Nanosatellites, https://satsearch.co/products/steel-electronique-ninano-on-board-computer-for-nanosatellites, (accessed 24.01.23).

[6] Wikipedia, DBSCAN, https://en.wikipedia.org/wiki/DBSCAN, (accessed 24.01.23).

[7] Wikipedia, XtratuM, https://en.wikipedia.org/wiki/XtratuM, (accessed 24.01.23).

[8] Digilent, ZYBO, https://digilent.com/shop/zybo-z7-zynq-7000-arm-fpga-soc-development-board/, (accessed 24.01.23).