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

Improving AI Monitoring of Early Life Satellites Using Transfer Learning

Audric Baron[a]*, Pierre-Baptiste Lambert[b], Pauline Delande[c], Jean-Yves Tourneret[d], Julien Lesouple[e], Serge Fabre[f]

[a] *Centre National d'Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France*, audric.baron@cnes.fr
[b] *Centre National d'Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France*, pierre-baptiste.lambert@cnes.fr
[c] *Centre National d'Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France*, pauline.delande@cnes.fr
[d] *Université de Toulouse, IRIT/ENSEEIHT/TéSA, 2 Rue Camichel, 31071 Toulouse, France*, jean-yves.tourneret@toulouse-inp.fr
[e] *Ecole Nationale de l'Aviation Civile (ENAC), 7 avenue Edouard Belin, 31400 Toulouse, France*, julien.lesouple@enac.fr
[f] *Laboratoire TéSA (Télécommunications Spatiales et Aéronautiques), 7 Boulevard de la Gare, 31500 Toulouse, France*, serge.fabre@tesa.prd.fr
* Corresponding Author

Abstract

In the last decades, many space domain actors such as the Centre National d'Etudes Spatiales (CNES) have begun to use Artificial Intelligence to monitor spacecraft housekeeping telemetry. These novel techniques are able to identify atypical behaviours and potential satellite anomalies that cannot be detected by more standard monitoring approaches. However, AI methods have an important drawback: they need a significant amount of data to be able to "learn" the nominal behaviour of a spacecraft and then detect novelties in new telemetry, which is not suitable for a satellite in the beginning of life where in-flight telemetry is very scarce. One way to bypass the scarcity of data is Transfer Learning (TL). Depending on the use case, operators may have already-available telemetry either from on-the-ground Assembly, Integration, and Test (AIT) of the spacecraft, from full-digital or hybrid simulators, or from in-flight telemetry of one or multiple "twin-spacecraft" in case of a constellation with already-launched units. This already-available telemetry is often close, but not perfectly similar, to in-flight telemetry of the newly-launched spacecraft to be monitored. The idea of TL is therefore to use this large and existing database (the source database), coupled with the first in-flight telemetry from the new spacecraft (the target database), to be able to mathematically-design a relevant AI learning model. In 2022, CNES and TéSA laboratory have worked together and have identified two TL methods to detect anomalies in telemetry of early life satellites with few data, by working directly on the telemetry dataset (the learning domain) or on the model learned from the target database. The first TL method consists in mathematically modifying the decision boundary estimated by a One-Class Support Vector Machine (OC-SVM) algorithm applied to the source database to match the target database. The second method based on "Domain Transfer" consists in building an "extended" learning domain made up with the relevant data from both the source and target databases, which is used to build a learning model. These two algorithms have been evaluated with real Earth Observation satellite telemetry. The preliminary outcomes of this research show promising results. Further work will consist in implementing these methods operationally so that AI monitoring methods can be used from the very beginning-of-life of CNES satellites. The main conclusion of this work is that TL can be an interesting tool to monitor spacecraft housekeeping telemetry during the first 6 months after the launch of a satellite.

Keywords: Artificial Intelligence, Machine Learning, Anomaly Detection, Satellite Telemetry Monitoring, One-Class Support Vector Machine

Nomenclature

B_1 : source database
B_2 : target database
$\boldsymbol{x}_{ij}$: ith vector of the database B_j
$\boldsymbol{n}_j$: number of vectors of the database B_j

Acronyms/Abbreviations

Assembly, Integration and Test (AIT)
Artificial Intelligence (AI)
Beginning of Life (BoL)
Centre National d'Etude Spatiales (CNES)
Machine Learning (ML)
One-Class Support Vector Machine (OC-SVM)
Principal Component Analysis (PCA)
Télécommunications Spatiales et Aéronautiques (TéSA)
Transfer Learning (TL)
Telemetry (TM)

1. Introduction

Detecting anomalies in spacecraft housekeeping telemetry (TM) is a challenge for many operators. Nowadays, there are several monitoring methods that can be used to detect unexpected and unusual behaviours in satellite telemetry. Unfortunately, even if these methods can provide interesting detection performance, standard monitoring approaches (such as Out-Of-Limit checks) are not always able to detect every atypical behaviour and potential satellite anomalies. As a consequence, many space domain actors such as CNES have begun to use Artificial Intelligence (AI) in addition to legacy methods to monitor their satellite fleets.

After conducting a proper state-of-art and comparing the performance of several existing Machine Learning (ML) mathematical methods, CNES has developed and tuned its own One-Class Support Vector Machine (OC-SVM) algorithm. This software, called NOSTRADAMUS (New Operational SofTwaRe for Automatic Detection of Anomalies based on Machine learning and Unsupervised feature Selection) [1], is used in CNES-operated missions since 2016 and has already demonstrated its capability to detect atypical and never seen behaviour in satellite telemetry. However, the main limitation of AI methods, such as NOSTRADAMUS, is that their learning step has to be performed on many nominal data from a spacecraft to detect correctly the anomalies affecting TM. The learning dataset of NOSTRADAMUS is currently tuned to require about a year of in-flight telemetry. The resulting amount of vectors has been empirically proven to provide a good trade-off between detection performance and computational complexity, even if good results might be obtained with less data in another operational context.

However, CNES has observed that AI-based techniques such as NOSTRADAMUS cannot always be used as-is in the beginning of life (BoL) of a new spacecraft until enough in-flight telemetry has been collected. From this observation, CNES decided to carry out a one-year study with the TéSA Laboratory in 2021-2022 in order to identify efficient Transfer Learning (TL) methods that would be relevant with the OC-SVM algorithm of NOSTRADAMUS in order to be implemented and tested with a dataset with a reduced volume of training data from a newly launched satellite. The idea of TL is to tune mathematically the learning model of a system, from another similar system that has a larger learning database. This study considers learning models using jointly TM from a newly launched satellite and TM from either another similar older satellite (for instance from the same constellation) with an already-significant amount of in-flight data, from Assembly, Integration and Test (AIT) or from simulations of the spacecraft. It aims at achieving a proper anomaly detection for a recently launched satellite with a limited quantity of acquired data, which would not be possible without the help of complementary data. In this paper, the so-called "source database" corresponds to the database that was acquired before the database of interest (from an in-flight comparable satellite for instance) while the "target database" is the database of the recently launched satellite. The target database is smaller than the source database but the two databases are related (for instance the design of the two considered satellites is very similar), which will be harnessed by TL.

This paper investigates two mathematical TL methods that have been considered for anomaly detection in TM, with a focus on the results and lessons learned for the operational application of TL. The first TL method consists in mathematically modifying the decision boundary obtained from a One-Class Support Vector Machine (OC-SVM) algorithm applied to the source data using the target database. The second method consists in constructing an "extended" learning domain containing the target database and the relevant source data having a structural similarity with the target database.

This paper is organised as follows: Section 2 summarizes the context of the study and details the data used by the NOSTRADAMUS software. Section 3 introduces the first TL method that consists in modifying the decision frontier resulting from the OC-SVM method by using data from the new learning database. Section 4 describes the second TL method based on the construction of an extended learning domain containing data from the source and target databases. The quality criterion used to evaluate the TL algorithms is defined in Section 5. Results obtained with the two TL algorithms when applied to satellite TM are presented and discussed in Section 6. Section 7 finally summarizes the different limits encountered during the study, and proposes potential applications for TL.

2. Context: NOSTRADAMUS

Since 2016, NOSTRADAMUS [1] is used within CNES operation centres for on-ground anomaly detection on spacecraft TM. Satellite TM consists of various continuous and discrete measurements resulting from its sensors. These measurements (called TM parameters) can be expressed in different units such as voltage, current, pressure, frequency, temperature, etc. Each TM parameter is therefore seen as a continuous stream of numerical values dumped and received each time the satellite passes over an Earth station. The purpose of NOSTRADAMUS is to detect the first weak signals due to an abnormal behaviour that might not be detected by standard monitoring and which could be an early warning sign of equipment failure.

2.1 Principle of NOSTRADAMUS

NOSTRADAMUS is an ML tool based on a OC-SVM algorithm encapsulated in an unsupervised and univariate method, operating in two modes referred to as **Learning Mode** and **Detection Mode**. The Learning Mode uses data corresponding to the nominal behaviour of a satellite in order to produce learning models. The Detection Mode uses satellite TM with a possible abnormal behaviour and compares it to the learning models resulting from the learning step to detect anomalies potential affecting these TM parameters (see Fig. 1).

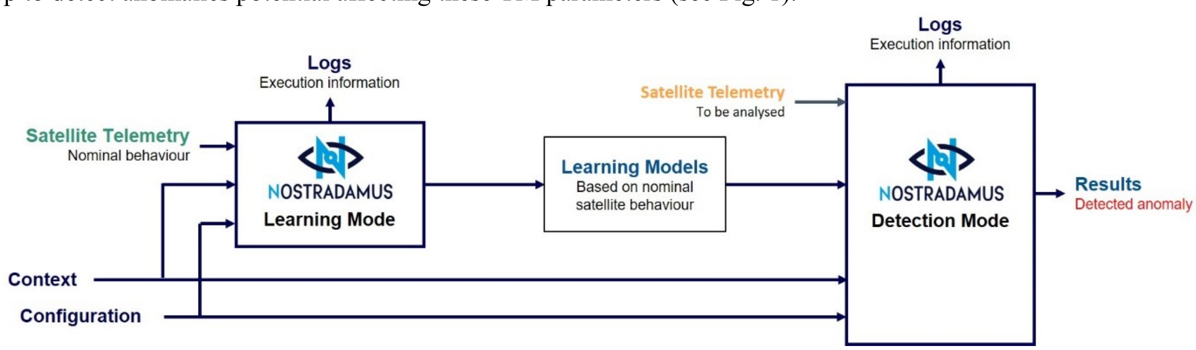

Fig. 1. NOSTRADAMUS high-level operation diagram.

The first steps of NOSTRADAMUS are to segment the data streams into time windows and to extract mathematical features in order to perform anomaly detection. The size of a window used in nominal operation for the segmentation of TM data is 24 hours. NOSTRADAMUS then converts each daily window of a TM parameter stream into a vector of 12 features (mean, maximum, minimum, standard deviation, skewness, kurtosis, energy and frequency features). These 12-dimensional vectors are then fed to a OC-SVM anomaly detection algorithm producing a separating boundary isolating the nominal data from the anomalies (the nominal data is contained inside the separating boundary whereas the anomalies are located outside this boundary).

2.1.1 Learning mode

The first step of the Learning Mode consists of injecting TM parameters associated with a nominal behaviour of a satellite to a OC-SVM algorithm. In a second step, the OC-SVM algorithm determines for each TM parameter a non-linear boundary – the decision boundary – that contains most of the data contained in the learning dataset. Traditionally around 300 time windows of length 24 hour (one year) are considered, which leads to 300 12D vectors of statistical features. The data vectors located inside this boundary are associated with the nominal behaviour of the satellite (see Fig. 2), whereas the vectors located outside the boundary are detected as anomalies. Once this decision boundary has been determined, a learning model has been generated. This learning model can be used to perform anomaly detection using a detection mode explained in the next section. Note that the user can adjust the maximum percentage of points located outside the boundary and the smoothness of this boundary.

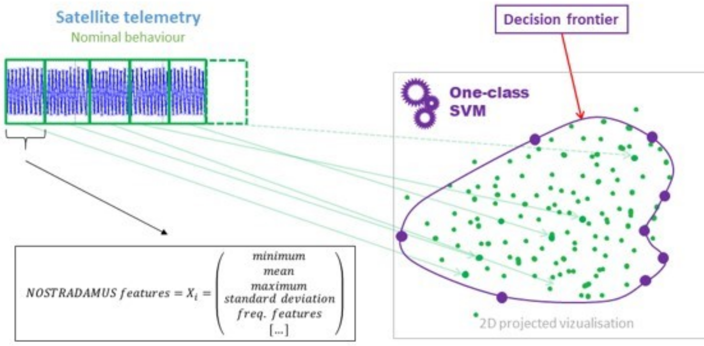

Fig. 2. NOSTRADAMUS learning mode.

2.1.2 Detection mode

The Detection Mode of NOSTRADAMUS, classifies the data inside or outside the decision boundary, thus performing anomaly detection. The data used for learning and detection should be ideally different. Typically, each week, an anomaly detection is performed on TM parameters from the seven previous days. For these TM parameters, any point inside the decision boundary is considered as nominal, whereas points located outside the boundary are considered as anomalies (see Fig. 3). Note that the distance between the boundary and a point located outside the nominal set can be used to define an anomaly score (ranging from 0 to 100) quantifying the probability that this point is an anomaly.

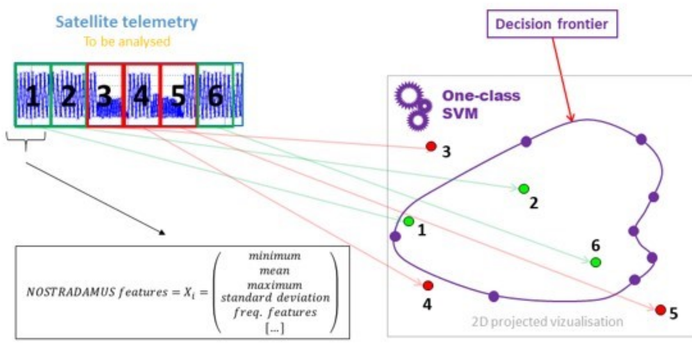

Fig. 3. NOSTRADAMUS detection mode.

2.2 Limits and interests

As described in Section 2.1 NOSTRADAMUS has been tuned to detect anomalies affecting TM of space missions monitored by CNES. However, the performance of NOSTRADAMUS decreases significantly in the BoL of a satellite. Indeed, the learning mode requires 12 months of data with nominal behaviour to ensure a good performance of anomaly detection, which is not available for a newly launched spacecraft. This inconvenience is difficult to circumvent, which has motivated this study related to TL for monitoring satellite TM.

3. Transfer Learning for OC-SVM

This section explains the principle of the TL method used with the OC-SVM algorithm to detect anomalies in satellite TM. Through this paper, the source and target databases are denoted as B_1 and B_2 respectively. The different elements of these databases are vectors. The ith vector of database B_1(resp. B_2) is denoted as $\boldsymbol{x}_{i1}$ (resp. $\boldsymbol{x}_{i2}$). The numbers of elements of B_1 and B_2 are denoted as n_1 and n_2.

3.1 Principle of Transfer Learning

The principle of the OC-SVM method is to find a separating hyperplane in a feature space defined as:

$$\boldsymbol{w}^T \phi(\boldsymbol{x}) - \rho = 0, \qquad (1)$$

where ϕ is a nonlinear transformation, $\boldsymbol{w}$ is a vector defining the separating hyperplane, ρ is a real offset and $\boldsymbol{x}$ is a vector from the database.

The separating hyperplane defined by (1) is determined in order to have most nominal data located inside its boundary (see Fig. 4. For an example where the separating boundary is displayed in red). After determining the separating hyperplane, the detection of anomalies can be performed as follows:

$$\text{Decide that } \boldsymbol{x} \text{ is not an anomaly if } \boldsymbol{w}^T\phi(\boldsymbol{x}) - \rho > 0, \quad (2)$$

$$\text{Decide that } \boldsymbol{x} \text{ is an anomaly if } \boldsymbol{w}^T\phi(\boldsymbol{x}) - \rho < 0. \quad (3)$$

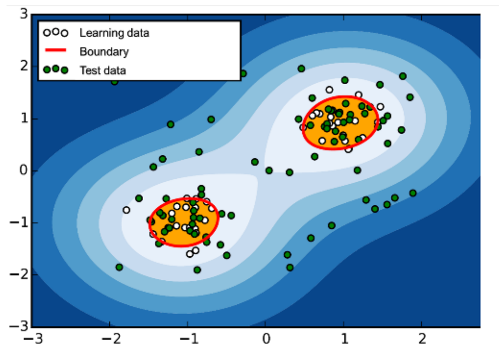

Fig. 4. Example in two dimensions of separating boundaries (figure taken from [2]).

The objective of TL is to detect anomalies in B_2 using B_1. A solution investigated in [3, 4] consists in transferring the separating function determined using the vectors $\boldsymbol{x}_{i1}$ (red points in Fig. 5) to the separating function that would be obtained with a larger number of data $\boldsymbol{x}_{i2}$, using only a small amount of data $\boldsymbol{x}_{i2}$ from B_2 and all the data $\boldsymbol{x}_{i1}$ from B_1.

In real-life applications where B_2 is growing with time, B_2 does not contain enough data in the BoL of the satellite to define a proper boundary for anomaly detection (the computed boundary obtained using a small amount of data is too far from the expected blue boundary displayed in Fig. 5). In this case, TL can be used to find a separating function that will be closer to the blue boundary of Fig. 5 (obtained using the OC-SVM method with a large number of data $\boldsymbol{x}_{i2}$ from B_2). Note that this study has been limited to two satellites even if the generalization to any number of satellites would be possible.

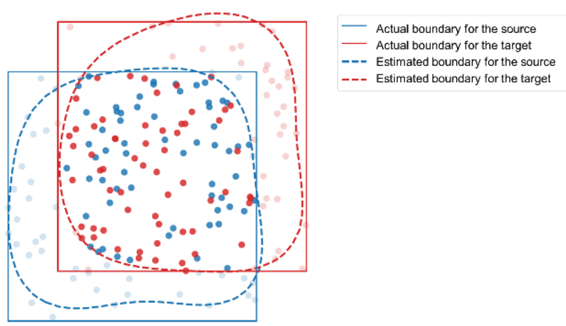

Fig. 5. Examples of boundaries estimated for two distinct databases.

The TL method used for the OC-SVM algorithm applied to NOSTRADAMUS is summarized in Fig. 6.

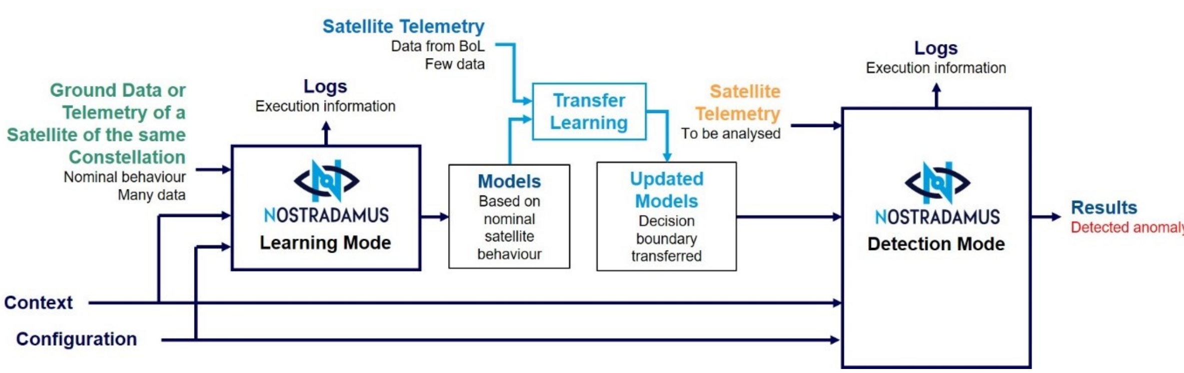

Fig. 6. Principles of TL for OC-SVM.

3.2 Formulation of the Transfer Learning problem

The method considered in this work has been first presented in [3, 4]. It involves a OC-SVM learning [5] on the (distinct) source and the target databases. The separating hyperplane of a given database is defined by two parts: a part common to the source and the target databases, and another part specific to the database. The balance between these two parts is done by the use of a hyperparameter μ, which balances the importance of the common part of the hyperplane with respect to the specific part of the hyperplane. The aim of this method is to find a hyperparameter μ such that the decision boundary is as close as possible as the boundary that would be obtained with enough data $\boldsymbol{x}_{i2}$. For this study, B_1 has been obtained using a first satellite already in orbit with a large amount of data, and B_2 has been obtained from a second satellite with a smaller amount of data, which is supposed to have similar characteristics than the first satellite.

Considering the equation of a separating hyperplane (1), the method of TL used in [3, 4] decomposes the vector normal to the separating hyperplane for the source $\boldsymbol{w}_1$ and the vector normal to the separating hyperplane for the target $\boldsymbol{w}_2$ as follows:

$$\boldsymbol{w}_1 = \mu \boldsymbol{w}_0 + (1-\mu)\boldsymbol{v}_1, \quad (4)$$

$$\boldsymbol{w}_2 = \mu \boldsymbol{w}_0 + (1-\mu)\boldsymbol{v}_2, \quad (5)$$

where $\boldsymbol{w}_0$ is the part common to the two databases, $\boldsymbol{v}_1$ is the part specific to the source, $\boldsymbol{v}_2$ is the part specific to the target, and $\mu \in [0,1]$ allows us to give more or less importance to the common part of the source and target databases. Note that for $\mu = 0$, the source database is not taken into account for the target learning, i.e., giving identical results to a standard OC-SVM, whereas for $\mu = 1$, all source data are considered, giving a solution identical to a OC-SVM algorithm trained using on the union of the source and target databases.

3.3 Automatic tuning of the parameter μ

The method proposed in [4] for estimating the hyperparameter μ provided too small values of μ, whereas μ should be closer to 1 for our application. The value considered in this study was obtained by cross validation, leading to $\mu = 0.8$. Some new methods for selecting the hyperparameter μ, that are more suited to the problem of detecting anomalies in CNES TM, have also been investigated and are presented below.

Sørensen-Dice index

An estimation of the similarity between the source and the target databases can be used to estimate the hyperparameter μ. To quantify this similarity, the Sørensen-Dice index is an interesting measure [6, 7] that considers the proportion of vectors declared as normal for the two sets $\boldsymbol{X}_1$ of the source and $\boldsymbol{X}_2$ of the target (see Fig. 5). More precisely, the Sørensen-Dice index between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is defined as [6, 7]:

$$s(\boldsymbol{X}_1, \boldsymbol{X}_2) = \frac{2|\boldsymbol{X}_1 \cap \boldsymbol{X}_2|}{|\boldsymbol{X}_1| + |\boldsymbol{X}_2|}, \quad (6)$$

where denotes $|A|$ denotes the cardinal of set A. The hyperparameter μ can then be estimated using the Sørensen-Dice index as:

$$\tilde{\mu} = \frac{s(\boldsymbol{X}_1, \boldsymbol{X}_2)}{\log(n_2)}, \quad (7)$$

where n_2 is the number of target data.

Correlation in the feature space

The second method that can be used to quantify the similarity between the sets $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is to evaluate the correlation of the data located inside the representation space. In the case of a Gaussian kernel (used in this work), the data is projected onto an infinite-dimensional unit sphere. A way to calculate the similarity between two datasets in this space is to average the scalar products between the transformed vectors $\phi(\boldsymbol{x}_{j1})$ and $\phi(\boldsymbol{x}_{j2})$ leading to the following estimator of μ:

$$\breve{\mu} = \frac{1}{\log(n_2)} \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \phi(\boldsymbol{x}_{i1})^T \phi(\boldsymbol{x}_{j2}). \quad (8)$$

4. Transfer Learning and domain transformation

TL can also be conducted by applying a domain transformation to an anomaly detection method as in [8]. The idea is to define an extended domain composed of the target database and part of the source database, and then to perform anomaly detection using this extended domain. The method suggested in [8] is based on the principle of k-nearest neighbours, which is limited to TL with a “semi-supervised anomaly detection method” referred to as SSKNNO. Since NOSTRADAMUS uses the OC-SVM algorithm, we have modified the algorithm of [8] to make it applicable to OC-SVM applied to the extended domain.

4.1 Principle of domain transformation

Domain transformation is based on the similarity between the vectors $\boldsymbol{x}_{i1}$ from the source domain B_1 and the vectors $\boldsymbol{x}_{i2}$ from the target domain B_2. The idea proposed in [8, 9] is to find the similarity between the vectors $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ with the following restrictions:

- The set B_1 contains some elements labelled as normal or anomaly and unlabelled elements (semi-supervised learning);
- The set B_2 contains only unlabelled elements.

In this study, B_1 is built using data from a satellite with labelled vectors (resulting for instance from the application of the OC-SVM algorithm), and B_2 is constituted using all unlabelled TM data from another satellite.

The method proposed in [8, 9] consists in defining an anomaly score for each element $\boldsymbol{x}_{i2}$ of B_2 using the neighbours of $\boldsymbol{x}_{i2}$ contained in an extended target domain D^*. Once this extended domain has been defined, a standard anomaly detection algorithm can be run on the elements of this extended domain. The learning mode of the OC-SVM algorithm, here NOSTRADAMUS, is run on this new domain allowing a learning model to be generated. The detection mode of NOSTRADAMUS is then used on the new satellite TM in order to detect anomalies. The principle of this method of TL based on domain transformation is summarized in Fig. 7.

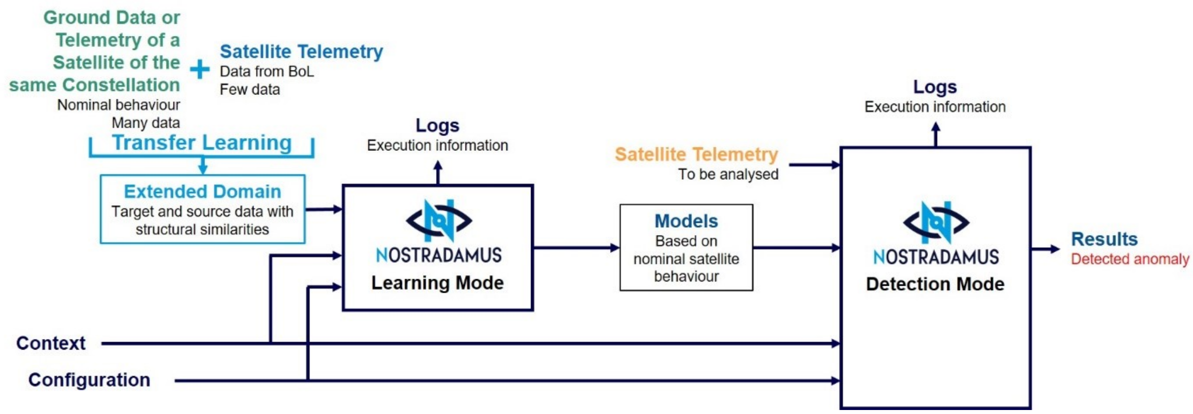

Fig. 7. TL by domain transformation.

4.1.1 Extended domain

The extended target domain, denoted as D^*, contains all the target vectors and some source elements transferred to the target domain. More precisely, an element from the source domain is transferred to the target domain only when its neighbours in the source domain and in the target domain have the same local structure. This local structure is measured by a distance d_1 between the barycentres of the neighbours in the source and target domains and a distance d_2 between the covariance matrices of the neighbours in the source and target domains. An example of construction of the extended domain (extracted from [10]) is displayed in Fig. 8.

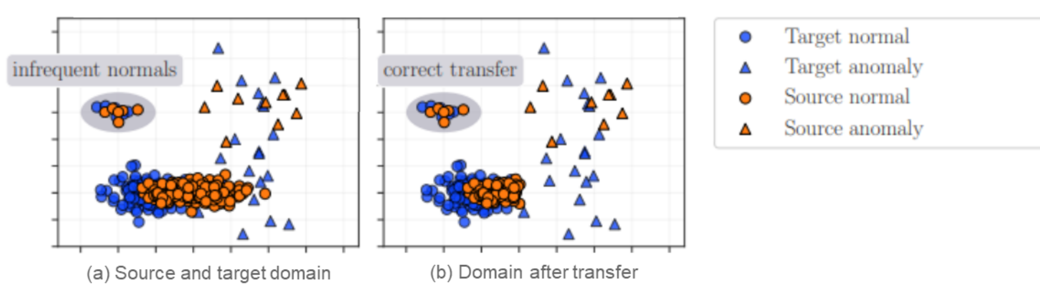

Fig. 8. Examples of data before transfer (a) and after transfer (b) (Figure extracted from [10]).

5. Quality criteria

The definition of quality criteria adapted to TL in the context of NOSTRADAMUS is essential to evaluate the performance of the different algorithms. In this study, two databases from two already launched satellites from the same constellation (therefore very similar in terms of design) have been used during training, more precisely the source database and the target database. The quantity of data provided by the source database has been chosen to ensure a good detection of the anomalies affecting the first satellite. After launching a new satellite (whose behaviour is supposed to be similar to the first satellite used for the creation of the source database), new data become available and the amount of data belonging to the target database increases with time. The challenge of TL is to design an anomaly detection algorithm using jointly the small quantity of data from the target database and the data from the source database in order to improve the detection of anomalies contained in the target database. In the very BoL of the satellite, a small part of the target database is used jointly with the source database to perform TL. A progressive increase of the size of the target database is considered to take into account the regular addition of data over time. The criteria used to evaluated the performance of the TL algorithm are based on classic performance measures used in detection, namely accuracy, precision, recall and F1 score, as explained in Sections 5.1 and 5.2.

5.1 Criterion using a part of the target database

The algorithm is first tested by extracting a part of the target database and by comparing the detection results obtained using this partial database with those obtained using the entire target database with the OC-SVM algorithm (NOSTRADAMUS). The principle of this strategy is illustrated in Fig. 9, where a part of the target database with P vectors and the entire source database with M vectors are used for training. The results of this TL, with P anomaly scores and P labels (nominal or anomaly) are compared to those obtained using the entire target database with N vectors from which the P vectors have been extracted for TL ($N > P$). Standard ML performance measures (Accuracy, Precision, Recall, F_1 Score, ROC AUC and Average precision) are used for comparison (see Appendix A for details). Note that the algorithm used for TL should be tuned as the standard OC-SVM algorithm used for anomaly detection.

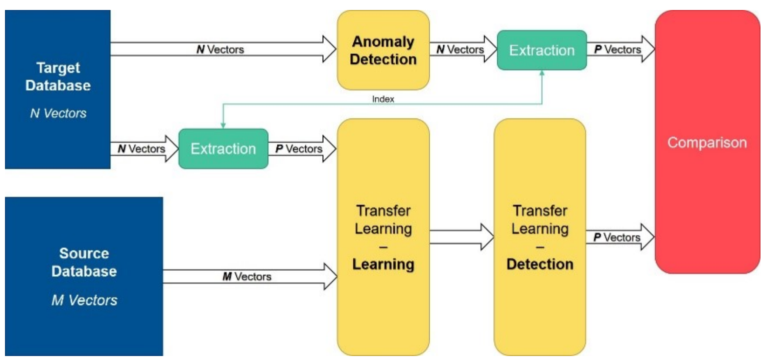

Fig. 9. Principle of the strategy used to evaluate the performance of TL using part of the source database.

5.2 Criterion using the entire target database

The order of appearance between all the vectors of the target database is lost when using a part of the target database as explained in Section 5.1, since the selected P vectors are chosen randomly. In order to avoid this drawback, it is interesting to define a criterion using the entire target database. The principle of this criterion is detailed in Fig. 10. Its purpose is to perform the learning mode of the TL algorithm on the extracted part of the target database and to perform an anomaly detection on the entire target database. The advantage of this method is to increase the number of vectors to be tested without losing the temporal order.

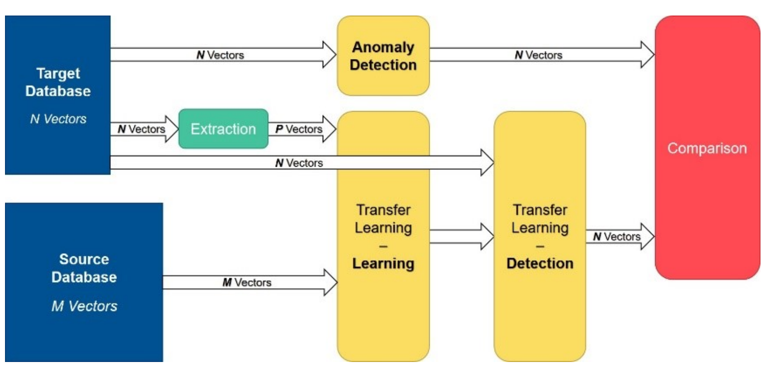

Fig. 10. Quality criterion using the entire target database.

6. Results

This section presents results obtained with real TM data from two CNES-operated satellites from the same constellation. One satellite is considered as the source database, denoted as "SAT1", and the other one denoted as the target database, "SAT2" is used for the target database. For this study, a total of 13 TM parameters from SAT1 and SAT2 have been tested. Among these 13 parameters, different satellite equipment of different type (temperature, current, velocity, frequency …) have been considered. The analysis has been conducted using sequences of one day of TM that have been split into 2 vectors (i.e., two time windows of 12-hour each), giving about 1000 vectors per database. Criteria I and II described in Sections 5.1 and 5.2 have been applied to all the tested parameters. In order to see the contribution of TL, the size of the extracted data from the target database has been increased gradually from 0 to 500 vectors (50% of the database).

Several algorithms using OC-SVM as a detector have been tested using these 13 parameters. One main algorithm has been developed by method, the "ocsvmtl" for the TL method mentioned in Section 3 and the "locIT" for Domain Transfer mentioned in Section 4. The performance of TL is analysed for these two types of algorithms by comparing their behaviour with the OC-SVM algorithm used by NOSTRADAMUS without TL. The algorithms using OC-SVM as a detector, mentioned in Sections 3 and 4, are summarized below:

- *OneClassSVM*: the OC-SVM algorithm of NOSTRADAMUS [1]
- *ocvsmtl*: the OC-SVM TL algorithm described in [4] with $\mu = 0.8$
- *ocvsmtl-sorensen*: the OC-SVM TL algorithm with μ tuned using the Sørensen–Dice index (7)
- *ocvsmtl-correl-kernel*: the OC-SVM TL algorithm with μ tuned using the correlation criterion in the kernel-induced representation space (8)
- *tesa_locIT_OneClassSVM*: the OC-SVM version of the domain transformation algorithm of Section 4.

As mentioned in Section 5.1, it is important to set the parameters for the anomaly detection part of the different algorithms in the same way in order to obtain a relevant comparison. Therefore, the parameters used in anomaly detection have the same values for all algorithms, chosen beforehand for NOSTRADAMUS according to the satellite data. Note that domain-transfer-based algorithm cannot accept a target database having less than 10 vectors, which explains why the corresponding curves start at 15 vectors.

6.1 Results obtained on satellite telemetry

In this section, only 5 of the 13 TM parameters are mention, relevant or not. The TM parameter #1 corresponds to the amplification of the signal received by the transponder. The TM parameter #2 corresponds to the measured current of the AOCS subsystem. The TM parameter #3 corresponds to the current of the tracking device. The TM parameter #4 corresponds to the measured temperature of the S-band receiver. The TM parameter #5 corresponds to the measured temperature of the S-band transmitter. Table *1* displays the Sørensen index defined in Section 3.3 for these five parameters.

Table 1. Sørensen–Dice index for parameters #1 to #5

	#1	#2	#3	#4	#5
Sørensen index	0.98	0.97	0.94	0.99	0.99

In order to evaluate the TL method applied to a database, a visualization of the parameters can be done using a principal component analysis (PCA), keeping the two and three principal components of the target satellite. Fig. 11 to Fig. 13 display a projection in two and three dimensions of both databases for parameters #1 to #3 (SAT1 in orange and SAT2 in blue). Fig. 11 shows a strong superimposition of the source and target databases for the parameter #1, also explained by a high Sørensen index (as for parameters #4 and #5); Fig. 12 shows also a significant intersection between the two databases but more spread for the parameter #2, whereas Fig. 13 shows a slight translation between the two databases with two spread clusters for the parameter #3, this explains why Sørensen index is lower. The first two figures correspond to two satellites that have a very close behaviour, whereas the second figure corresponds to two satellite producing shifted data. Indeed, the two satellites have very similar characteristics for the equipment of Fig. 11 and Fig. 12, and the current of the equipment displayed in Fig. 13 has an offset between the two satellites because it needs to be on longer than on the other satellite. These examples typically illustrate that parameters between two satellites of the same constellation can have many similarities and some differences. Therefore, TL has to be able to perform anomaly detection based on these similarities while accounting for these differences.

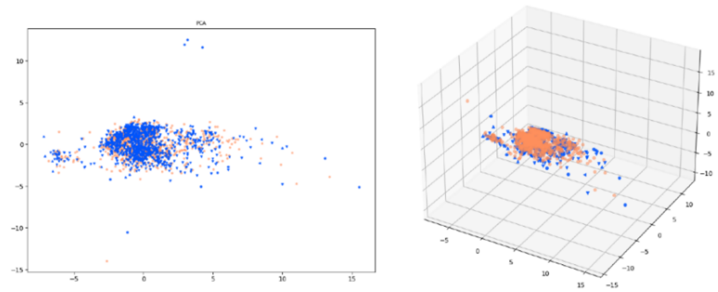

Fig. 11. Spatial projection in the PCA of SAT2, for the first 2 components (on the left) and for the first 3 components (on the right) for the TM parameter #1 (SAT1 in orange, SAT2 in blue).

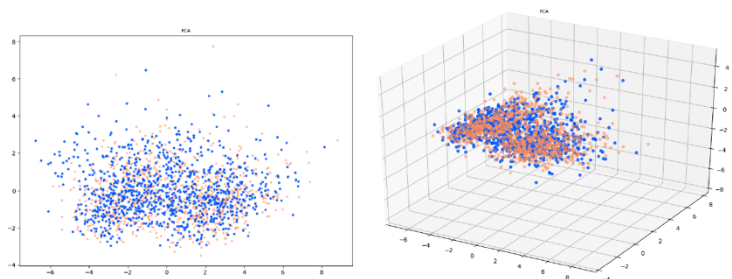

Fig. 12. Spatial projection in the PCA of SAT2, for the first 2 components (on the left) and for the first 3 components (on the right) for the TM parameter #2 (SAT1 in orange, SAT2 in blue).

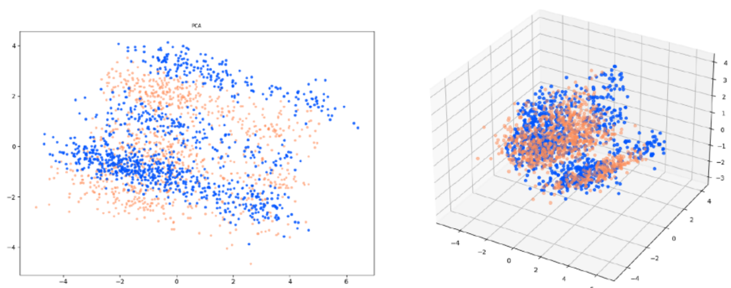

Fig. 13. Spatial projection in the PCA of SAT2, for the first 2 components (on the left) and for the first 3 components (on the right) for the TM parameter #3 (SAT1 in orange, SAT2 in blue).

For each criterion (I and II), several metrics have been calculated depending on the size of the database. These metrics are described in Appendix A. The corresponding figures depicted below represent, for each OC-SVM algorithm, a comparison of the results of the TL with the result obtained by using NOSTRADAMUS applied to the entire target database (which is the objective that we try to obtain using TL).

Fig. 14 to Fig. 18 display the evolution of the F_1 score, generally observed for most parameters, for algorithms based on OC-SVM for criteria I and II. Note that the F_1 score is the most significant metrics for understanding the performance of the tested algorithms since it is based on precision and recall. More precisely, Fig. 14 and Fig. 15 present results for two parameters (denoted as parameters #3 and #4) in terms of F_1 score for the different algorithms using criterion I. Fig. 14 shows that TL has improved the performance of anomaly detection, while Fig. 15 shows that the proposed algorithm has a good performance that is closer to that obtained using NOSTRADAMUS. Note that the behaviour of most of the tested parameters is similar to that of Fig. 14.

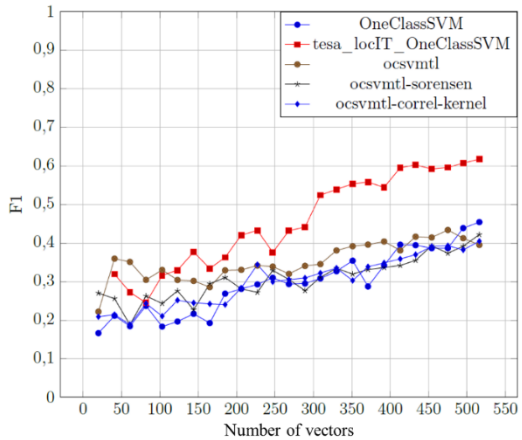

Fig. 14. Performance of the different TL methods using the OC-SVM method in terms of F_1 with criterion I for the TM parameter #4.

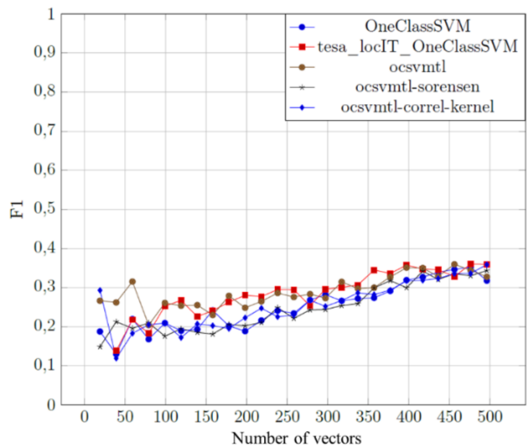

Fig. 15. Performance of the different TL methods using the OC-SVM method in terms of F_1 with criterion I for the TM parameter #2.

These results associated with criterion I lead to the following comments:

- Fig. 14 shows that the *tesa_locIT_OneClassSVM* algorithm has a significant improved performance when compared to the other algorithms and starts to be advantageous from 100 vectors, and is 18% more efficient than NOSTRADAMUS at 300 vectors. The *ocsvmtl* algorithm with $\mu = 0.8$ outperforms the other algorithms but is less efficient than *tesa_locIT_OneClassSVM*. Note that the *ocsvmtl* algorithm is efficient from 0 to 150 vectors. This parameter is interesting because some anomalies may be detected when a non-routine behaviour occurs on the transponder itself or nearby equipment. Its dataset contains several anomalies that the TL algorithm manages to detect quickly. Parameters #1 and #5 have the same shape.
- Fig. 15 shows that the *tesa_locIT_OneClassSVM* algorithm has also interesting results for a number of vectors of the target database higher than 100. The *ocsvmtl* algorithm with $\mu = 0.8$ outperforms the other algorithms but is almost as efficient as *tesa_locIT_OneClassSVM*. Note that the *ocsvmtl* algorithm is the most efficient from 0 to 70 vectors, around 11% more efficient that NOSTRADAMUS. The gain on this parameter does not seem as important as on the temperature above. Parameter #3 has the same shape.

The *tesa_locIT_OneClassSVM* algorithm performs better when the intersection between the source and the target databases is high (for a high Sørensen index).

Note also that the other *ocsvmtl* algorithms are not interesting when compared to the reference algorithm because the methods used for the calibration of μ tend to underestimate its optimal value.

Fig. 16 and Fig. 17 present results associated with two parameters in terms of F_1 score for criterion II. Both figures show that the TL algorithm has improved the performance of anomaly detection.

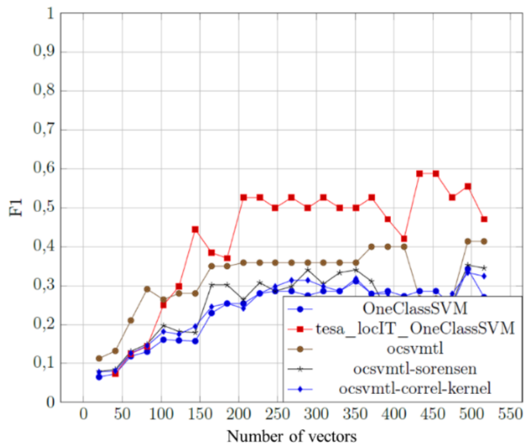

Fig. 16. Performance of the different TL methods using the OC-SVM method in terms of F_1 with criterion II for the TM parameter #5.

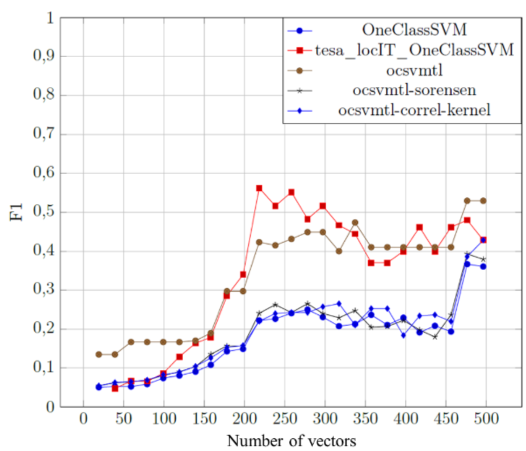

Fig. 17. Performances of the different TL methods using the OC-SVM method in terms of F_1 with criterion II for the TM parameter #3.

These figures for criterion II lead to the following comments:

- Fig. 16 shows that the *tesa_locIT_OneClassSVM* algorithm has a significant improved performance when compared to the others algorithms and starts to be advantageous from 100 vectors. At 200 vectors, it is 25% more efficient than NOSTRADAMUS. Moreover, the *ocsvmtl* algorithm with $\mu = 0.8$ performs better than the other algorithms for a number of vectors in the target lower than 100, with 10% more efficiency. This parameter is also interesting as Fig. 14. Similar shapes are obtained for parameters #1 and #5.
- Fig. 17 shows that the *tesa_locIT_OneClassSVM* algorithm provides also interesting results beyond 550 vectors. The *ocsvmtl* algorithm with $\mu = 0.8$ performs better than all the other algorithms especially for a small number of vectors in the target database (here below 150 vectors) and still challenges the *tesa_locIT_OneClassSVM* algorithm. At 200 vectors, the *tesa_locIT_OneClassSVM* and the *ocsvmtl* algorithms are around 15% more efficient than NOSTRADAMUS. This parameter is the same that of Fig. 13. For this current, TL is more interesting than the current of Fig. 15. For these parameters (#2 and #3), the *tesa_locIT_OneClassSVM* and the *ocsvmtl* algorithms have similar performance and curves are close.

Fig. 18 shows the detection results in terms of true positive (TP), true negative (TN), false positive (FP) and false negative (FN), for the different algorithms based on OC-SVM using criterion II depending on the size of the target database for the parameter #1, as Table 2 but for two different sizes of the target database . A total of 11 anomalies are present in this database. The *ocsvmtl* algorithm reaches quickly to a good number of detection, while the *tesa_locIT_OneClassSVM* algorithm later achieves the same number of good detections but with a lower rate of false detection. Recall illustrates that NOSTRADAMUS reaches the same detections only after 400 vectors for this parameter, and then demonstrates that these TL algorithms are able to perform anomaly detection with few vectors.

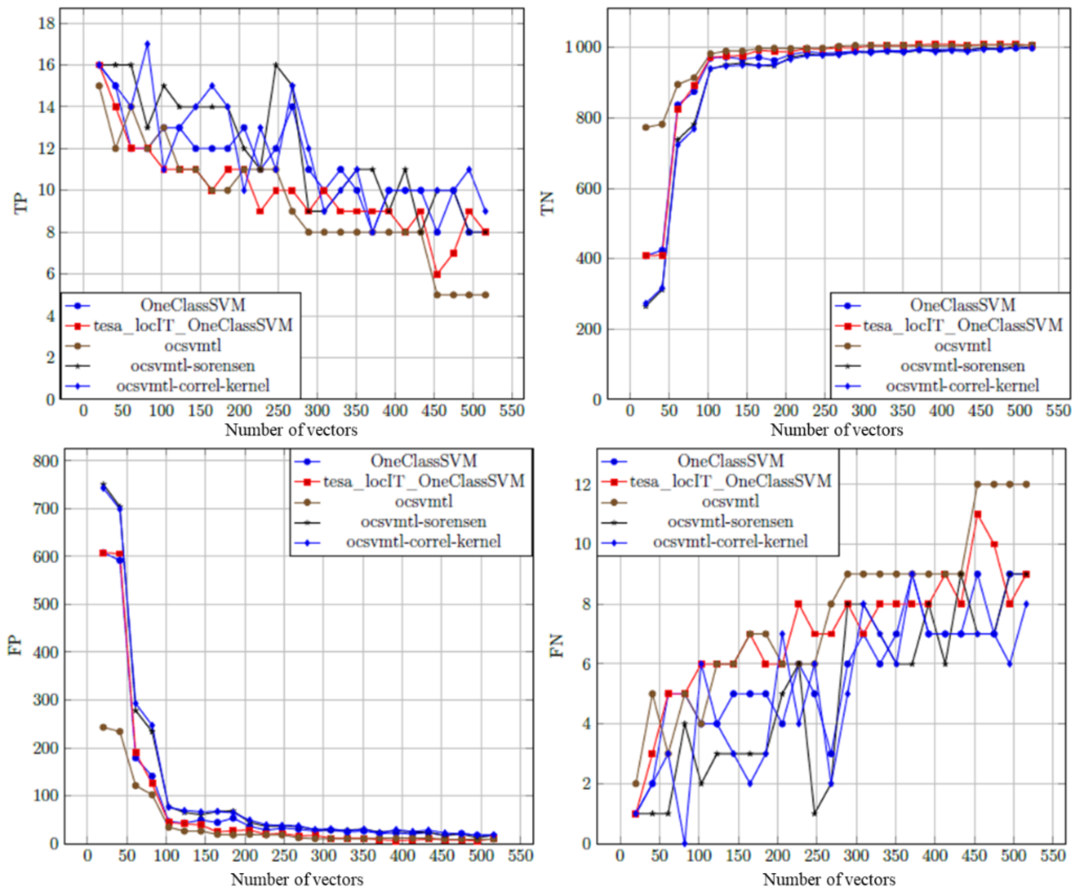

Fig. 18. Performance of the different TL methods using the OC-SVM method in terms of TP, TN, FP, FN with criterion II for the TM parameter #1.

Table 2. TL results for TM parameter #1

	TP	TN	FP	FN
tesa_locIT_OneClassSVM (150 vectors)	13	980	36	4
tesa_locIT_OneClassSVM (350 vectors)	10	1008	8	7
ocsvmtl-μ = 0:8 (150 vectors)	10	996	20	7
ocsvmtl-μ = 0:8 (350 vectors)	11	1005	11	6
ocsvmtl-correl-kernel (150 vectors)	14	949	67	3
ocsvmtl-correl-kernel (350 vectors)	1033	991	25	8
ocsvmtl-sorensen (150 vectors)	14	950	66	3
ocsvmtl-sorensen (350 vectors)	9	992	24	8

6.2 Analyses of results

This section presents the global analyses of the results obtained on all the TM parameters with the quality criteria described in Section 5.

6.2.1 Criterion I

The *tesa_locIT_OneClassSVM* algorithm outperforms the other algorithms for several parameters (7 parameters out of 13). The efficiency of this algorithm is directly linked to the intersection between the source and the target databases, this superimposition is measured between 0 and 1 with the Sørensen–Dice index. The algorithm starts to be advantageous from 50 to 150 vectors for 6 of the parameters considered in this work. It results in an increase in precision and to a lesser extent in recall, compared to the reference OC-SVM method. It means that the number of false positives tends to be lower with *tesa_locIT_OneClassSVM* when compared to the other methods.

Concerning the *ocsvmtl* type algorithms, for 7 parameters out of 13, the algorithm run with a fixed value $\mu = 0.8$ provides better results than the others, while being less efficient than the *tesa_locIT_OneClassSVM* algorithm. For these parameters, TL with this algorithm is efficient from 0 to 150 vectors. Generally, accuracy is improved when using TL.

TL with the other algorithms seems to be less efficient than *tesa_locIT_OneClassSVM*. However, it starts to be efficient earlier, with a very small number of vectors in the target database. The other *ocsvmtl* algorithms have a limited advantage with respect to the reference algorithm. Indeed, the proposed methods for calibrating μ tend to underestimate its optimal value, and rather quickly prefer low values of μ, which does not support TL. On the contrary, the method with a value of μ equal to 0.8 favours the transfer in all cases.

6.2.2 Criterion II

This criterion has the advantage of taking into account the order with which the data appear. The *tesa_locIT_OneClassSVM* algorithm systematically presents advantages, except for one parameter. TL is advantageous from 100 or 150 vectors to the end for 11 parameters, and between 100 and 350 vectors for one of these parameters. The precision is always improved while the recall is often deteriorated, when compared to the other algorithms.

For lower values of the number of vectors in the target (< 50), the *ocsvmtl* algorithm with $\mu = 0.8$ performs better than the other algorithms.

6.2.2.1 Improvement and tuning of tesa_locIT_OneClassSVM

As mentioned above, the *tesa_locIT_OneClassSVM* algorithm is generally the most efficient when the target database contains between 300 and 500 vectors. However, the final target boundary for a low number of target vectors is too close to the source boundary and thus its performance is reduced when the number of target vectors is too low. For practical operations, we propose to use the *locIT* algorithm as defined in Section 4, only if the number of target vectors is above $P = 300$ (where 300 is an empirical value below which *locIT* performance is poor). If $P < 300$, an extended domain is constituted with source and target data, where the number of source vectors allows this extended domain to have a size close to the size of the full target database. When additional target data are acquired, the contribution of the source will decrease.

The performance of this improved version of the *tesa_locIT_OneClassSVM* algorithm is better for 11 out of 13 parameters. As an example, Fig. 19 illustrates a better value of F_1 score for TM parameter #1.

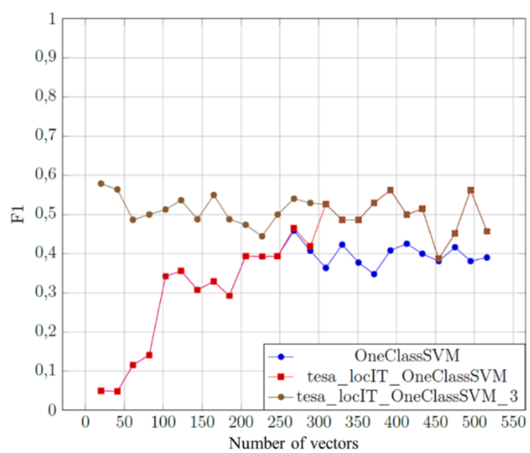

Fig. 19. Performance of the *tesa_locIT_OneClassSVM* methods and its improved version in terms of F_1 with criterion II for the TM parameter #1.

7. Discussion

To summarise, the first method is based on an intersection on the two databases, and the second on the average of the scalar products between the source and the target in the representation space generated by the kernel. However, through the study of several analysed parameters, it has been observed that these methods tended to favour low values of μ for small target databases, avoiding to take advantage of the source database, which is the interest of TL.

Another version of the *ocsvmtl* algorithm with a parabolic interpolation seems promising since the first days and could be more developed further. Indeed, the first results show that the parabolic tuning have good performance between 0 and 250 vectors.

Another algorithm, the kNNO algorithm, has also been tested during this study, and is more efficient for performing a good anomaly detection when few vectors are used for TL (even better than the SSKNNO algorithm based on domain transformation). However, it is important to note that for this algorithm, learning and detection cannot be done jointly, which can be a problem in space operations where both modes are essential.

Most of the results have been obtained by comparison of performance between the TL algorithm and NOSTRADAMUS. Although the performance of the TL algorithms generally seems better than NOSTRADAMUS without TL at the BoL, this is not enough to justify that they are very efficient and give very good results. The labelled database of this study was not large enough to conclude on the performance of the algorithms.

Moreover, these TL algorithms have been applied to only 13 TM parameters from two already existing satellites. Indeed, these tests could better estimate the performance of the TL algorithms with more than 13 parameters.

It could be interesting to test them with simulated or AIT data. It could be useful in the case where the first satellite of a constellation is launched, for which in-flight TM does not already exist. It is conceivable that the simulated/AIT data is less similar to in-flight TM. The “transfer” (of data or the modification of the learning boundary) is less obvious because the data is less comparable, then the behaviour and the efficiency of TL methods should be studied when they are used for anomaly detection since the BoL.

8. Conclusions

Several Transfer Learning methods can be used to improve telemetry monitoring at the beginning of life of a satellite, as soon as some external data similar to the data of this satellite are available. During this study, two main Transfer Learning algorithms have been investigated: *ocsvmtl* based on the transfer of the separating boundary via a One-Class Support Vector Machine, and *tesa_locIT_OneClassSVM* based on domain transformation.

Our analyses have demonstrated that the second algorithm provides better performance, and is effective from the very beginning of Transfer Learning and up to about 250 days of data. Beyond this limit, it is preferable to use the One-Class SVM algorithm of reference without Transfer Learning. Regarding the first algorithm *ocsvmtl*, it is effective from the start and up to 125 days of data. After acquiring 125 days of telemetry data, it is recommended to use the standard One-Class SVM algorithm without Transfer Learning.

According to our results and with the parameters tested, Transfer Learning can improve the performance of anomaly detection until 30% compared to NOSTRADAMUS alone, but this comparison with the quality criteria does not provide performance metrics for Transfer Learning. However, 13 parameters are not enough to fully decide on the usefulness of the Transfer Learning and to go further, it would be necessary to launch tests on more parameters. Indeed, it could be interesting to operationally test this process on a space mission ae the beginning of its life in order to see if it meets the need with all parameters of this mission. An interesting space mission for this application could be the swarms of satellites, which would produce a significant reduction in monitoring time.

Acknowledgements

The authors would like to thank the CNES operations AI team and the TéSA laboratory for their support. They are also very grateful to Pierre Beauseroy from Troyes University of Technologies for fruitful exchanges related to the algorithms developed in [3] and [4].

Appendix A (Machine Learning Performance Measures)

A TL algorithms return two types of data:

- binary values (in nominal situation or in anomalies), called “labels”;
- continuous values (which can be probabilities of anomalies or distances to boundaries depending on the algorithm used), called “scores”.

Performance metrics can be divided into two groups: group #1 compares labels with scores and group #2 compares labels only.

A.1 Label/label comparison

This paper has used the following parameters:

- True Positive (TP): an anomaly detected as an anomaly
- False Positive (FP): a nominal case detected as an anomaly
- True Negative (TN): a nominal case detected as a nominal case
- False Negative (FN): an anomaly detected as a nominal case

Accuracy

This performance measure is computed using the *accuracy_score* function of the Scikit-Learn (sklearn) library, which indicates the percentage of correct detections:

$$A = \frac{TP + TN}{Total}, \qquad (9)$$

where *Total* is the total number of data in the learning database.

Precision

This performance measure is computed using the *precision_score* function of the sklearn library. The precision is the model’s ability to detect an anomaly when this anomaly is present. It is the number of true positives divided by the total number of detections:

$$P = \frac{TP}{TP + FP}. \qquad (10)$$

Recall

This performance measure is computed using the *recall_score* function of the sklearn library. The recall is the algorithm’s ability to detect all anomalies:

$$R = \frac{TP}{TP + FN}. \qquad (11)$$

F_1 Score

This performance measure is computed suing the *f1_score* function of the sklearn library. Criterion F_1 combines the two previous criteria and is defined as the product of precision and recall normalised by the arithmetic mean of these two criteria:

$$F_1 = \frac{2 \times P \times R}{P + R}. \qquad (12)$$

A.2 Label/score comparison

ROC AUC

This performance measure is computed using the *roc_auc_score* function of the sklearn library. The ROC AUC criterion (area under the curve “Receiver Operational Characteristics”) is calculated as the area under the curve expressing the true positive rate (TPR) as a function of the false positive rate (FPR) for different decision thresholds. The higher ROCAUC, the better the detector, as illustrated in Fig. 20.

Fig. 20. ROC curve

Average Precision

This performance measure is computed using the *average_precision_score* function of the sklearn library. Average precision indicates whether a model can correctly identify all positive examples (defects) without accidentally marking too many negative examples (nominal case) as positive. Therefore, the average precision is high when the model can correctly handle positives.

The mean precision is calculated as the area under a curve that expresses precision as a function of recall for different decision thresholds. This criterion is more suitable than ROC AUC when the classes are very unbalanced, which is the case for the detection of anomalies (because there are few anomalies in the database).

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, in Proc. 14th Int. Conf. on Space Operations, Daejeon, Korea, 2016.

[2] J.-Y. Tourneret, L. Chaari, A. Ferrari, and C. Richard, CNES technical report, Analyse TM pour diagnostics, Jul. 2015.

[3] Y. Xue, Dynamic Transfer Learning for One-class Classification: a Multi-task Learning Approach, PhD thesis, University of Troyes, France, 2018.

[4] Y. Xue and P. Beauseroy, Transfer learning for one class SVM adaptation to limited data distribution change, Pattern Recognit. Lett., vol. 100, pp. 177–123, 2017.

[5] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput., vol. 13, no. 7, pp. 1443–1471, 2001.

[6] T. A. Sorensen, A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons, Biol. Skar., vol. 5, pp. 1–34, 1948.

[7] L. R. Dice, Measures of the amount of ecologic association between species, Ecology, vol. 26, no. 3, pp. 297–302, 1945.

[8] V. Vercruyssen, W. Meert, and J. Davis, Transfer learning for anomaly detection through localized and unsupervised instance selection, Proc. AAAI Conf. on Artificial Intelligence, New-York, USA, vol. 34, no. 4, 2020.

[9] V. Vercruyssen, Designing Anomaly Detection Algorithms that Exploit Flexible Supervision, PhD Thesis, KU Leuven, Belgium, 2020.

[10] J.-Y. Tourneret, J. Lesouple, and S. Fabre, CNES technical report, Transfer Learning appliqué au monitoring de télémesures satellites, R-S21/BS-0003-076, 2022.