

AN UNSUPERVISED LEARNING-BASED MANOEUVRE DETECTION METHOD FOR RESIDENT SPACE OBJECT PATTERN OF LIFE CHARACTERISATION

R. Cipollone⁽¹⁾, E. Raviola⁽²⁾, and P. Di Lizia⁽³⁾

⁽¹⁾*Department of Aerospace Science and Technology, Politecnico di Milano via Giuseppe La Masa 32 20156 Milano, Italy, Email: riccardo.cipollone@polimi.it*

⁽²⁾*Department of Aerospace Science and Technology, Politecnico di Milano via Giuseppe La Masa 32 20156 Milano, Italy, Email: enrico.raviola@mail.polimi.it*

⁽³⁾*Department of Aerospace Science and Technology, Politecnico di Milano via Giuseppe La Masa 32 20156 Milano, Italy, Email: pierluigi.dilizia@polimi.it*

ABSTRACT

The problem revolving around the status of the near-Earth environment is complex and ever-changing, requiring a collective effort to monitor the evolution of the resident space object population from both institutions and industry. Reaching a comprehensive Space Situational Awareness has thus become of utmost importance, due to the immediate effects on every phase of a space mission, from its design to the operations foreseen after its deployment. A fundamental aspect involved in the surveillance and tracking of space objects is taking operational ones into account in terms of their manoeuvring capabilities. Actively controlled objects can, in fact, hamper the correlation of new tracks, leading to false or missed associations, and reduce orbit determination accuracy, introducing unmodeled contributions to the assumed dynamics. One way of addressing the problem consists in characterising the pattern of life of operative objects in terms of manoeuvres in order to understand how likely it is for a target to be actively controlled within a specific time frame. The main showstopper identified in the literature is the shortage of manoeuvre histories used to generate labels for supervised learning algorithms. This severely reduces a method's generalization capability. The current work aims to overcome this limitation by resorting to unsupervised learning techniques commonly used for anomaly detection purposes to identify manoeuvres with no prior information on the target's control history via a sequential autoencoder. The proposed application considers ballistic motion as the 'normal behaviour' while identifying manoeuvring arcs as out-of-nominal samples.

Keywords: Maneuver detection; Space Surveillance and Tracking; Machine Learning; Autoencoder.

1. INTRODUCTION

With the wide availability of services for humanity and scientific research that rely on near-Earth space, the number of space objects, both operational and debris, has significantly grown in recent years. A key actor in this framework is represented by satellite constellations, particularly focusing on the Low Earth Orbit (LEO) region to provide everyday large-scale services, such as a robust Internet connection. These space objects are usually able to manoeuvre routinely to ensure proper orbit maintenance and collision avoidance. Given that manoeuvres plans are often not available to the public, space traffic becomes increasingly complex to manage and coordinate. Preserving space safety is, in fact, one of the most challenging and critical current objectives of the space technology field, as evidenced by the presence of various programs aimed at ensuring complete and responsive Space Situational Awareness (SSA). One of the most effective ways to achieve this is represented by Space Surveillance and Tracking (SST) activities for Resident Space Object (RSO) catalogue maintenance. Among them, the accurate and timely recognition of maneuvers performed by uncooperative objects in orbit around Earth is key to supporting the increasing population of active targets. An inevitable consequence of this kind of activities is that large amounts of structured data at different refinement stages are stored as by-product of SST measurement processing pipelines. This tendency has allowed data-driven algorithms, and machine learning in particular, to play a significant role in this branch of research, enabling the automation of several steps composing the detection and tracking process. This category of methods comes into play whenever model-based techniques are not viable to describe a phenomenon, the reason often being some key information is missing (i.e. anomaly detection, manoeuvre estimation), or whenever they are not convenient in terms of computational costs. This work investigates the former case, elaborating on how machine learning techniques can be applied to detect active control from a target orbital history. Leveraging the target's

past behaviour provides critical context for assessing its maneuvering probability within a given time frame. Several studies have performed statistical analyses on orbital data histories to identify events indicative of maneuvers. Traditional approaches often rely on statistical techniques and combinations or transformations of Keplerian osculating elements, using deviations from a locally fitted curve or their time derivatives as maneuver detection metrics [20][14]. However, these methods typically require case-dependent threshold tuning, necessitating continuous monitoring. An alternative approach proposed in [17] attempts to address this limitation through automated threshold adaptation. Machine learning techniques try to address this very limitation, emerging as a promising solution by leveraging large datasets of cataloged RSO information to encode maneuver-related variations and patterns for classification. Both unsupervised learning methods, such as clustering based on variation magnitudes [3], [19], and supervised techniques, including Support Vector Machines and Random Forests [25], [7], as well as deep learning architectures like Convolutional [6], [23] and Recurrent Neural Networks (RNNs) [5], have demonstrated significant potential. Notably, RNNs show promising results when their predictive capabilities are integrated with detection or classification mechanisms to anticipate maneuvers within a specified time horizon. In contrast to standard neural networks handling a single sample of features per training iteration, RNNs have the capability to take sequentiality into account via a hidden state, allowing them to process and interpret input time series more effectively. In order to determine which RNN variant best fits maneuver detection, an example of model comparison is reported in [1], assessing LSTM, Bi-LSTM, and GRU-based architectures in terms of control action identification from Cassini’s trajectory. In general, a significant limitation to this set of methods is the fact that they are based on labeled data, meaning that the history of past manoeuvring events must be available for them to start a training process. This is not always the case, and while some manoeuvre databases are available online (such as the International Laser Ranging System (ILRS)¹) they cover a restrained subset of targets. Therefore, a lot of trust is put on the model capability to generalise on such complex and regime-dependent phenomenon. A workaround to this shortcoming comes from a different approach to the topic stemming from anomaly detection literature. As stated in [18], there are several criteria which the identification of out-of-nominal behaviour can be based on, from clustering and density estimation to reconstruction and distance-based techniques. Among them, reconstruction error can surely benefit from data-driven models trained on a given past nominal behaviour, and autoencoders in particular prove to be a suitable solution for a wide range of tasks [8] [29]. An application of this concept to manoeuvre detection is reported in [13], where this architecture is exploited to recognize manoeuvres as anomalies in orbital data. The main limitation characterising this class of methods is the choice for a threshold to flag a sample as

anomalous. Building on this research direction, this work explores the use of a Long Short-Term Memory (LSTM) autoencoder, specifically designed to process sequential data and capture temporal dependencies as a data-driven way of reconstructing the ballistic motion of a target. A manoeuvre is identified whenever an anomaly is detected. The combination of data-driven modelling and unsupervised thresholding, and the decoupling between training and testing targets (a debris and an active satellite respectively) represent the main contributions of the work. The analysis reported in this paper is constrained to typical maneuvers performed by Low Earth Orbit (LEO) objects, primarily aiming at orbit-raising and plane-preservation. This focus is driven by the limited availability of publicly accessible maneuvering records, sourced from the International Laser Ranging System (ILRS) database, with which the proposed method is tested.

2. FUNDAMENTALS

This section details the mathematical principles and tools that are employed to build the proposed manoeuvre detection method, starting from a description of the TLE data format to the unsupervised learning architecture chosen for the application.

2.1. Two-Line Element Set

The TLE is a data format consisting of orbital elements and information about an RSO at a given epoch. It usually consists of two 69-character lines of data that can be properly processed to extract the orbital state (position and velocity) of the target. A typical TLE (Time and Location Ephemeris) consists of two lines of data, each containing 69 characters. These lines can be processed to calculate the RSO’s mean orbital parameters in the geocentric coordinate system True Equator Mean Equinox (TEME). Additionally, this file format provides the value of B^* (BSTAR), which is an adjusted value of the ballistic coefficient based on the reference atmospheric density value, at one Earth radius. It’s important to note that B^* is not a physical quantity and typically includes all contributions that are not accurately modeled by the dynamics. As reported in [27], TLE files originate from an Orbit Determination (OD) process that utilizes observations from various sensors belonging to a known RSO. These observations are typically collected multiple times daily to update the target orbit in the catalog. This process necessitates the use of a specific propagator and a dynamical model selection to propagate a reference state, which encapsulates the latest orbital information about the object, to the observation epochs. Among the catalog maintenance activities, the Simplified General Perturbations-4 (SGP4) [12][28] is the most commonly employed propagator.

¹<https://ilrs.gsfc.nasa.gov/>, last accessed: 2024-02-09

2.2. Manoeuvres and Orbital parameters

Any adjustment to the orbit that is due to active control can be achieved by altering a satellite's velocity [26]. As a result, analysing changes in orbital parameters can provide insights on the occurrence of maneuvers, and if there are noticeable variations in the orbital elements across time, a detection can be performed. Following a perturbed orbital dynamics [26], when a sudden velocity change $\Delta \mathbf{v}$ is applied at any firing point during a near-circular orbit, it simultaneously alters the orbital elements of the satellite, as reported in the following set of equations.

$$\begin{cases} \Delta a = \frac{2}{n} \Delta v_r \\ \Delta e = \frac{1}{na} (\sin f \Delta v_r + 2 \cos f \Delta v_u) \\ \Delta i = \frac{1}{na} \cos(\omega + f) \Delta v_h \\ \Delta \Omega = \frac{1}{na} \frac{\sin(\omega + f)}{\sin i} \Delta v_h \\ \Delta \omega = \frac{1}{na} (-\cos f \Delta v_r + 2 \sin f \Delta v_u - \Delta \Omega \cos i) \end{cases} \quad (1)$$

Where a , e , i , ω , Ω represent, in order, semi-major axis, eccentricity, inclination, argument of pericentre and the right ascension of the ascending node; f stands for the right ascension point along the orbit in terms of true anomaly while n is the mean motion, and Δv_u , Δv_r , Δv_h denote the impulsive velocity in the transverse, radial, and normal directions, respectively.

2.3. Machine Learning Algorithms

When dealing with large, structured datasets characterized by complex feature interactions and intricate distributions, machine learning provides an effective approach for extracting underlying patterns and insights. Most machine learning techniques rely on an optimization process, commonly referred to as training, where a model's parameters are iteratively adjusted to best fit the available data. Once trained on a sufficiently representative dataset, the model can generalize and infer information about unseen data from the same distribution.

This work specifically focuses on the application of Neural Networks (NN), a class of machine learning models that construct modular functions composed of interconnected neurons organized in layers to approximate a given data distribution. While neural networks are primarily used for regression and classification, they enable a broad range of advanced tasks, from computer vision to anomaly detection and error prediction. At their core, neural networks operate through individual neurons (shown in Fig.1), which apply a weighted linear transformation to the input, followed by a non-linear activation function, expressed as $g(\mathbf{W}\mathbf{x} + \mathbf{b})$. These neurons are arranged into layers, each capturing features at varying levels of abstraction. As deeper architectures allow

for better modeling of complex, non-linear patterns, they require increased computational resources, placing them within the domain of Deep Learning.

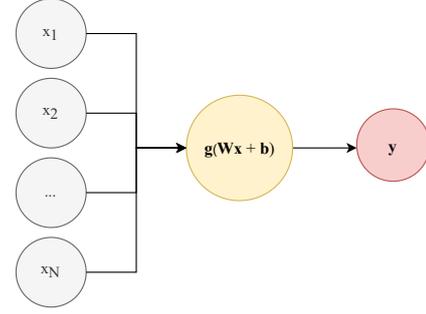


Figure 1. The figure shows a basic neuron architecture, mapping a linear combination of weights w_i and input features x_i through a non-linear activation function g

Given the sequential nature of orbital history data, this study focuses on Recurrent Neural Networks (RNNs), a specialized class of neural networks designed to capture temporal dependencies by maintaining a hidden state that propagates information through time [11]. RNNs process input sequences by updating their hidden state \mathbf{h}_t based on both the current input \mathbf{x}_t and the hidden state \mathbf{h}_{t-1} from the previous time step. RNNs are known to have one main limitation concerning their inability to encode long-term time dependencies due to vanishing errors used to build the loss function during back-propagation across time steps. With their information content fading progressively, the associated training times increase. A widely used solution to this issue is represented by the LSTM cell, an alternative basic unit to build an RNN, enabling selective preservation of the long and short-term information content that affects the current time step by using different gates (with corresponding weights and biases) and a memory cell \mathbf{c} to filter and modulate the information according to its source and distance in time [10]. The internal structure of the LSTM cell is made up of 3 gates with different roles, interacting with each other:

- the input gate \mathbf{i}_t , taking information from the previous hidden state \mathbf{h}_{t-1} and current input \mathbf{x}_t ;
- the output gate \mathbf{o}_t , with a similar structure but independent weight and bias terms;
- forget gate \mathbf{f}_t , modulating information according to hidden state distance in time.

The above-mentioned sequential model is thus used as building block for an autoencoder. This architecture consists of a specific structure designed for unsupervised learning tasks. It aims at two main objectives: encoding the input data into a concise, understandable latent representation, and then decode it to restore the original input. Autoencoders are thus trained to identify hidden

combination of variables derived from the input data that effectively represent the distribution of the data, but are not readily observable. This is what the latent space is: a summarised representation composed of a collection of latent variables tailored to a particular input dataset. Throughout the training process, the autoencoder gains knowledge about which latent variables work best for accurately recreating the original data, following the typical optimisation of a specific cost, usually the Mean Squared Error (MSE) loss function. Only the most effective details are thereby filtered from the original input by this latent space representation. Despite differing in code dimension, number of layers, number of nodes and the loss function, all autoencoder variants share the core structural components displayed in Fig.2.

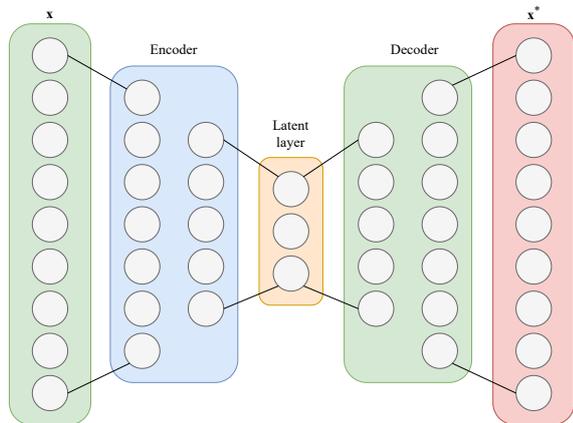


Figure 2. The scheme shows the symmetrical architecture of an autoencoder.

The application for which this network is used in the proposed technique focuses on its reconstruction capabilities so that a reliable check on anomalies can be set up by analysing the reconstruction error magnitude. This means that a threshold has to be selected to distinguish between anomalous and nominal data. In this case, this is done by means of an unsupervised clustering algorithm, namely the K-Means. It is an unsupervised machine learning algorithm designed to divide a dataset into distinct groups, or clusters, based on the similarities among data points [9]. The algorithm aims to group points such that those within the same cluster are more alike than those in different clusters. This approach is especially effective for uncovering hidden patterns or structures within the data.

The K-Means algorithm begins by selecting a fixed number of clusters, denoted by k . The main steps of the process are as follows:

1. k initial centroids (or centers) are chosen, either randomly or based on specific criteria, to represent the starting point for each cluster.
2. every data point is allocated to the nearest centroid,

typically measured by Euclidean distance, consequently grouping points.

3. for each cluster, a new centroid is computed by finding the mean of all points assigned to that cluster. This new point is used as updated centroid in the next iteration.
4. the algorithm repeats steps 2 and 3 until the centroids coordinates do not change significantly or a maximum number of iterations is reached.

The final output is a partition of the dataset into k clusters, each represented by a centroid that captures the average or center of the points within that cluster.

3. DATA AND ALGORITHM

This section provides a thorough description of the data employed to train and deploy the manoeuvre detection models together with the processing pipeline involved to achieve the results.

3.1. Data analysis and pre-processing

To identify spacecraft manoeuvres based on their unlabeled Pattern of Life, a preliminary analysis of the orbital data extracted from TLEs is conducted. The objective is to gain a preliminary understanding of the types of manoeuvres performed and to identify correlation between changes in orbital parameters and maneuver events. The selected Keplerian elements for the study are a , e , i , ω , chosen to provide comprehensive coverage of impulses both in-plane and out-of-plane. Moreover, an unconventional analysis of the B^* parameter extracted from the TLE data reveals the (expected) correlation between periods of heightened solar activity during solar cycles 24 and 25 [2] and increased B^* values, as Fig. 3 shows.

This relationship underscores the role of B^* as an indicator of aerodynamic drag within the SGP4 orbital propagation model but rather it is able to reflect the environmental disturbances affecting an object. In the case at hand it hints at the necessity for a satellite to perform manoeuvres more or less frequently to maintain its orbit, particularly in LEO. For this reason it is included among the features on which to train the network.

The data are retrieved from TLEs as a set of dictionaries containing lists of epochs covering the observed period, analysed orbital elements, and Cartesian states for each object. A filtering phase is first carried out after extracting orbital data from the TLEs. It consists of discarding the Launch and Early Orbit Phase (LEOP) of the target's history and then ensuring that duplicate values corresponding to the same orbit determination process (usually the latter being a correction of the former [17]) are removed from the dataset. Next, the data sequence is

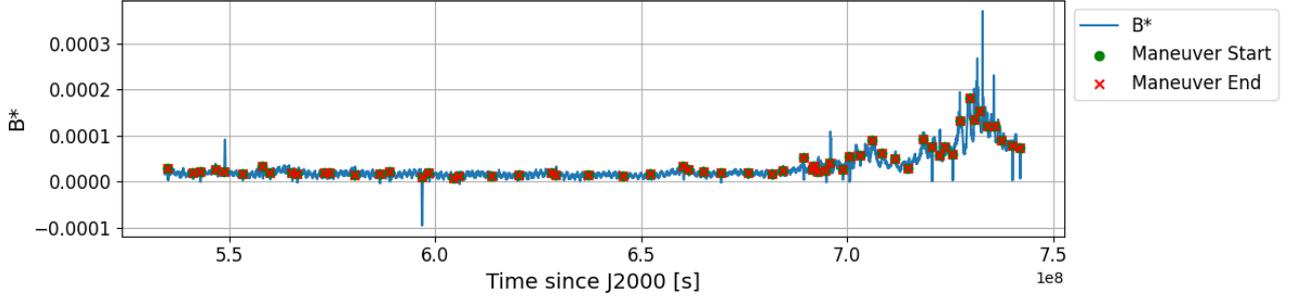


Figure 3. Sen3A B^* evolution

split into training, test, and validation sets, following a 70/20/10% rule. Each subset is then scaled via a Standard Scaler [21], ensuring that the features exhibited by every parameter span comparable ranges of values. The final step of data pre-processing reshapes the data into finite-size sequences and organize them into a 3-D format. This enables the neural network to learn from the temporal patterns within the data, which is crucial for detecting maneuvers and anomalies in orbital parameters. The sequence length is set at 6, based on a trial and error process, as this length has to balance the need for accurate anomaly detection and the ability of the network layers to preserve sequential information. Each sequence is thus defined by 6 consecutive sets of elements, that overlap for the most part across the entire dataset, following a moving window algorithms with unitary stride. The input shape is $(N, 6, 5)$, where N represents the total number of sequences, 6 is the number of samples per sequence and 5 corresponds to the number of selected features a, e, i, ω, B^* .

3.2. Network Design and Training

The procedure proposed for this work involves training a Bidirectional LSTM (Bi-LSTM) autoencoder on a target set of nominal data, where the RSO under analysis is expected to exhibit a purely ballistic behaviour. This allows the model to learn how to accurately reconstruct the input data as far as no active control is applied to the target. The selected neural network design to accomplish the task is based on Bi-LSTM layers, allowing for the model to encode the same sequence both forwards and backwards in time by doubling the weights associated to a typical LSTM cell [24]. A single architecture, reported in Tab.1, has proven to be a good fit for every scenario composing the testing campaign.

As for the training process, it is based on the optimisation of an MSE loss function, minimising the residuals between the reconstructed input and the actual one, as reported in the following equation:

$$MSE = \frac{1}{N \cdot T \cdot F} \sum_{n=1}^N \sum_{t=1}^T \sum_{f=1}^F (X_{n,t,f} - \hat{X}_{n,t,f})^2 \quad (2)$$

Layer	Output Shape
InputLayer	(None, WindowSize, Features)
Bi-LSTM	(None, WindowSize, 16)
Bi-LSTM	(None, 8)
z	(None, 8)
RepeatVector	(None, WindowSize, 8)
Bi-LSTM	(None, WindowSize, 8)
Bi-LSTM	(None, WindowSize, 16)
OutputLayer	(None, WindowSize, Features)

Table 1. Bi-LSTM autoencoder model. None represents the dynamic dimension of a batch

Where $X_{n,t,f}$ is the original input sample, $\hat{X}_{n,t,f}$ is the one reconstructed by the autoencoder, and $N, T,$ and F are respectively the number of sequences, the number of samples per sequence, and the number of considered features.

The optimisation algorithm selected for the training phase is Adam [15], representing the state of the art in terms of gradient descent-based methods, with an exponential decay law set on its learning rate.

3.3. Testing and Detection

Once the training phase is concluded, the models are tested on data that contain a mix of manoeuvring and ballistic trends to assess the presence of anomalies in the reconstruction error of the only semi-major axis. Since the neural network is tasked with reconstructing the multivariate input sequences, the reconstruction error is measured for each sequence and for each feature as a MSE:

$$AS = \frac{1}{T} \sum_{t=1}^T (X_t - \hat{X}_t)^2 \quad (3)$$

where X_t is the original sample, \hat{X}_t is the reconstructed one, and T is the number of samples per sequence. AS is therefore defined as Anomaly Score associated with the central epoch of the specific sequence on which the MSE is calculated, providing a figure for each time sample and

feature, and allowing for a temporal analysis of the reconstruction error's evolution. By examining the Anomaly Score profiles for each test performed, their characteristic jagged patterns are observed, with peaks corresponding to intervals where the network struggles to reconstruct the input. Among the compared curves, the one describing the semi-major axis is selected as the basis for a subsequent data post-processing procedure. This is because of the established one-to-one relationship between peaks and maneuvers, making it evident even to the naked eye the difference between normal data and anomalies. To verify the detection capability of the autoencoder model and to automate this process, thresholding is applied to the AS curve referring to this parameter.

When it comes to the detection procedure adopted there are 2 scenarios that have been applied to the same set of known RSOs.

The first one consists in a semi-supervised scenario, exploiting objects with publicly available manoeuvres to cherry-pick ballistic data between them and train the network to recognise that kind of behaviour as the nominal one. Once this is done, during the testing phase of the algorithm, the original version of the dataset (including manoeuvring arcs) is fed to the trained model in order to understand whether active control generates peaks in the reconstruction error trend. In this case, the model is trained with data coming from an active satellite, Sentinel-3A (Sen-3A)m while the testing phase is performed on both Sen3A and Sentinel-3B (Sen3B) data to preliminarily assess the network's generalisation capabilities as well. In order to automatically classify the reconstruction error sequence of the semi-major axis, a threshold is set across the entire test distribution, defined as three standard deviations around the mean AS value ($\mu + 3\sigma$) of the training set (assumed nominal), corresponding to a 99.7% confidence level on detecting anomalies. This simplified scenario is conceived to assess the performance of the autoencoder architecture selected as well as the effectiveness of the anomaly score formulation.

The second scenario, instead, represents the operative, unsupervised version of the technique. In order to make the training of the autoencoder independent of prior knowledge about the occurrence of manoeuvres throughout a target's history, a set of debris is selected with orbits that are as similar as possible to some pre-selected targets so as to have a training set featuring purely ballistic motion only. The test set is thus performed on the active satellites of interest in a way to obtain a background AS related to the orbital difference and peaks correlated to manoeuvring events. In this case, a set of quasi-polar debris from the NOAA16, NOAA17, CZ-4, CZ-4B, CZ-4C, Delta1, DMSP-5D and Fengyun-1C missions is selected to train the model. The trained network is then tested on Sen3A, Sen3B, Cryosat-2, Envisat and Saral. This time, to guarantee anomaly classification, a different approach is required compared to the previously described scenario. This is because the values of orbital parameters used during training (related to debris) are significantly different from those used to perform testing (belonging

to active satellites), and the fixed threshold defined by a σ factor-based method only would be too low to effectively separate anomalies. To address this issue, a two-step approach is employed. First, a 1D K-Means clustering algorithm is applied to the each test set to characterise the AS (the reconstruction errors on the semi-major axis) corresponding to nominal data. This selection is performed by identifying the most populous cluster among those that partition the set of AS . The reconstruction errors are intended to be divided into 2 up to 4 clusters, in accordance with the expected range of maneuver intensities that the satellites under analysis can reach. Once the nominal data are isolated, the σ factor thresholding technique is applied to the extracted distribution, and a boundary between nominal and anomalous is finally established.

4. RESULTS

This section delves into the details of the models performance assessment, describing the scenarios employed to establish how well the model is able to detect manoeuvres, starting from the semi-supervised scenario to the unsupervised one. The presented results are based on objects whose manoeuvres are publicly available, so by knowing the ground truth epochs, the detection results can be evaluated with the typical metrics used for classification tasks, being Precision (P), Recall (R) and F1 Score, defined as follow:

$$P = \frac{TP}{TP + FP} \quad (4)$$

$$R = \frac{TP}{TP + FN} \quad (5)$$

$$F1 = \frac{2PR}{P + R} \quad (6)$$

Where TP , FP and FN are true positives and false positives and false negatives respectively.

4.1. Semi-supervised Manoeuvre Detection

Tab.2 summarises the results of the network configuration associated to the first scenario of Sec.3.3. In this case, the model is trained on ballistic-only Sen3A data and tested on both the original Sen3A and Sen3B data sequence. The classification metrics values reported prove a performance level that is deemed compatible with the application, showing also encouraging results in terms of generalisation.

By examining the peaks in Sen3A semi-major axis Anomaly Score profiles caused by maneuvers displayed in Fig. 4, it is clear how sensitive the model is to any phenomenon that is not included in the ballistic subset of data used during the training process. The spikes are well-defined and localized, distinctly separated from nominal

	Precision	Recall	F1 Score
Sen3A	0.9565	1.000	0.9778
Sen3B	0.9200	0.9857	0.9517

Table 2. Semi-supervised method classification metrics

data, meaning that the adopted σ -based thresholding represents a suitable solution.

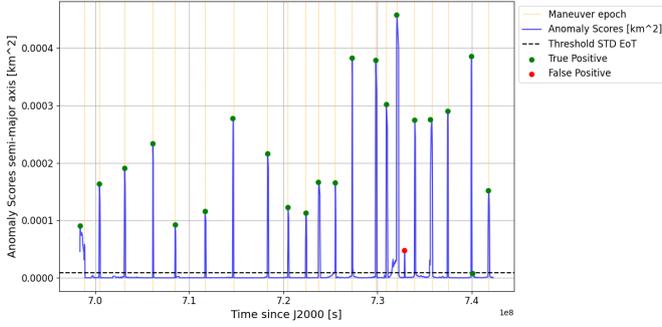


Figure 4. Sen3A semi-major axis Anomaly Scores and classification

In a preliminary attempt to characterise various types of maneuvers, at least distinguishing between in-plane and out-of-plane ones, a comparison is proposed between the AS of the semi-major axis and inclination, reported in Fig.5. In this specific case, the approach proves effective as the peaks in the inclination AS curve can clearly provide contextual information whenever the detected maneuvers involve the out-of-plane direction.

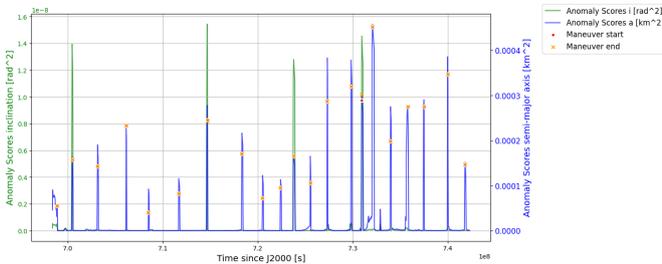


Figure 5. Sen3A semi-major axis Anomaly Scores and classification

4.2. Unsupervised Manoeuvre Detection

The results of the model referring to the second scenario in Sec.3.3, trained on a set of debris characterised by orbits that are similar to the target ones prove to achieve performance that are comparable to the semi-supervised case when tested with Sen3A and Sen3B. This holds regardless of the number of clusters selected to carry out the post-processing phased on the semi-major axis AS and the consequent thresholding applied to detect anomalies. Looking at the example in Fig. 6, where the reconstruction errors are split into 4 clusters, the clear separation of

nominal data (in blue) from the others is evident, clearly helping the subsequent thresholding step in being as sensitive and precise as possible.

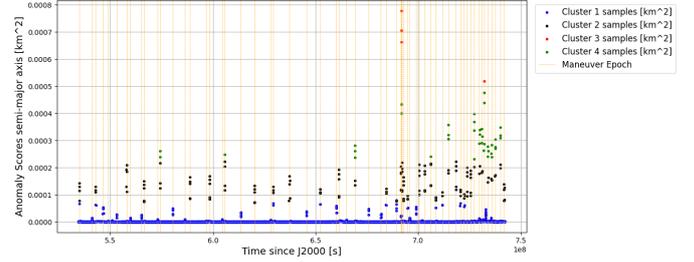


Figure 6. Sen3A 1D K-Means clustering clusters=4

The classification metric values for the Sen3A and Sen3B tests are reported in Tab.3, showing results that are deemed satisfactory for the application.

	Precision	Recall	F1 Score
Sen3A	0.9825	0.9333	0.9573
Sen3B	0.9718	0.9583	0.9650

Table 3. Unsupervised method classification metrics

Moving on to the tests conducted on Cryosat2 and Envisat, maneuver detection capabilities are highly influenced by the quality of the parameters extracted from TLE data. These profiles exhibit a different maneuver pattern and greater noise compared to the Sentinel satellites. As a result, the maneuvers are less distinguishable from the ballistic data. Thus, given this restrained level of resolution in the manoeuvre-related peaks in the semi-major axis AS trend, Selecting an optimal number of cluster becomes crucial to achieve a suitable thresholding process on the reconstruction error.

	Precision	Recall	F1 Score
clusters=2	1	0.6311	0.7738
clusters=3	0.9667	0.8447	0.9016
clusters=4	0.9438	0.8485	0.8936

Table 4. Cryosat2 unsupervised method classification metrics

The results in Tab. 4 demonstrate the model's ability to detect maneuvers for Cryosat2, with the best performance achieved by subdividing the reconstruction errors into 3 clusters. As shown in Tab.5, anomaly detection for Envisat is significantly worse compared to previous tests, and the model cannot be considered reliable. This is primarily due to the noise and significant irregularity characterizing the evolution of the input parameters, as well as a wider range of maneuver intensities. To highlight the importance and the impact of input data quality on both data reconstruction and subsequent anomaly detection, an additional example is provided using test data from the Saral satellite. In this case, as shown in Fig.7, there is a clear and localized transition in the extracted parameters going from noisy to smooth.

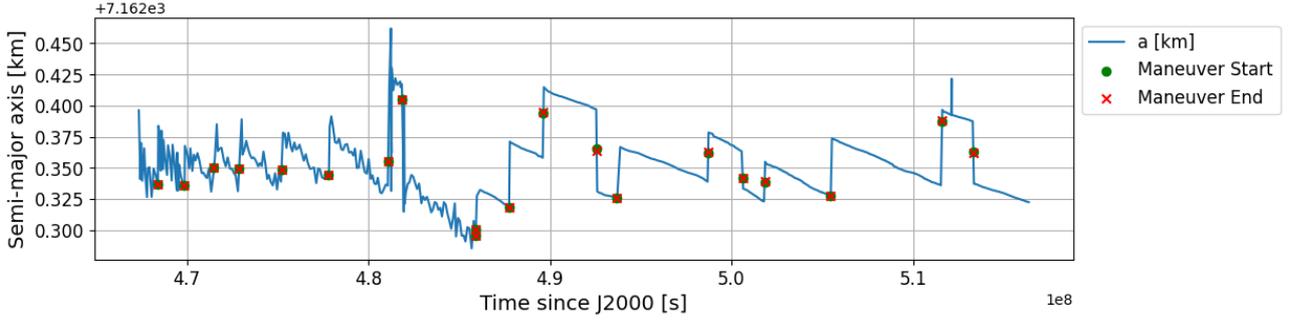


Figure 7. Saral semi-major axis

	Precision	Recall	F1 Score
clusters=2	0.8333	0.3529	0.4958
clusters=3	0.7273	0.4494	0.5555
clusters=4	0.6835	0.6353	0.6585

Table 5. Envisat unsupervised method classification metrics

This shift results in a corresponding variation in the ability of the model to reconstruct normal data, as shown in Fig.8.

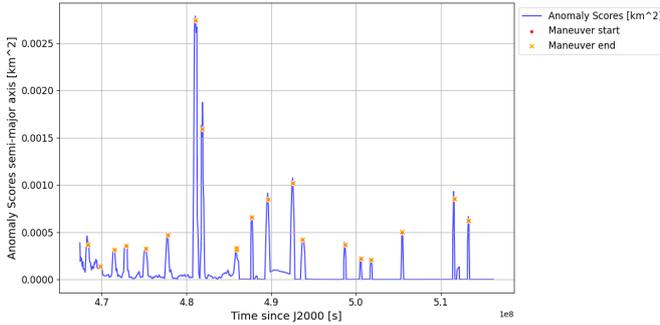


Figure 8. Saral semi-major axis Anomaly Scores

5. CONCLUSIONS AND FUTURE DEVELOPMENTS

The work described in this paper represents a first step towards unsupervised manoeuvre identification, showing how plugging clustering algorithms to the autoencoder architecture can lead to an almost completely autonomous detection pipeline. The orbital data selected as feature vector, $[a, e, i, \omega, B^*]$ prove to be well suited for training the autoencoder, and the choice of short temporal sequences as input proves useful in identifying maneuvers, distinguishing them from nominal data. The semi-major axis, due to its step-like behavior when it comes to LEO objects (mostly performing orbit maintenance manoeuvres), proves as the most effective parameter for detecting active control by analysing the reconstruction error trend. Meanwhile, inclination can complement it for

a preliminary manoeuvre characterisation assessments. As for the the semi-supervised testing campaign, employing Sen3A and Sen3B satellites, the main outcome consists in the fact that the Bi-LSTM autoencoders, trained on ballistic data from operational satellites, perform exceptionally well in telling manoeuvres apart from nominal data. Moreover, when paired with the reconstruction error post-processing step via clustering-based approaches, training the autoencoder with data from debris of a family of similar orbits proves effective to detect manoeuvres as anomalies of a purely ballistic behaviour. As far as satellites belonging to the same orbital family are concerned, the outcomes on the proposed methodology generalisation capabilities are encouraging. Nevertheless, employing Keplerian element profiles obtained from TLE data that contains errors or noise results in reconstruction inaccuracies.

With these considerations, it is possible to identify ideas for future developments. One area to work on is the dependency of the results on the quality of the input data: a possible workaround could be to investigate how noise filtering techniques can interact with the rest of the pre-processing pipeline in a way that the profiles do not lose their distinguishing features. Additionally, it would be useful to extend the number of satellites with known maneuvers for the testing phase, evaluating the effectiveness of the method even in cases involving objects from different orbital regimes, implying different maneuver patterns as well. Finally, an alternative advancement that would change the way the autoencoder architecture is exploited, could be to leverage the feature extraction capability of its encoder section and train it on a large dataset, to use the resulting latent space as input for a classification-aimed neural network.

ACKNOWLEDGEMENTS

This work is part of the results of the research activity funded by the agreement n. 2023-37-HH.0 for the project “Attività tecnico-scientifiche di supporto a C-SSA/ISOC e simulazione di architetture di sensori per SST”, established between the Italian Space Agency (ASI) and Politecnico di Milano (PoliMi).

REFERENCES

1. ALDabbas, Ashraf, and Zoltan Gal. "Recurrent neural network variants based model for Cassini-Huygens spacecraft trajectory modifications recognition." *Neural Computing and Applications* 34.16 (2022): 13575-13598.
2. Ashruf, Ayisha M., Ankush Bhaskar, and Tarun Kumar Pant. "Deciphering Solar Cycle Influence on Long-Term Orbital Deterioration of Space Debris in LEO and MEO orbits." *arXiv preprint arXiv:2405.08837* (2024).
3. Bai, X., Liao, C., Pan, X., Xu, M. : Mining Two-Line Element Data to Detect Orbital Maneuver for Satellite. *IEEE Access* 7 (2019).
4. Bank, Dor, Noam Koenigstein, and Raja Giryes. "Autoencoders." *Machine learning for data science handbook: data mining and knowledge discovery handbook* (2023): 353-374.
5. Cipollone, Riccardo, et al. "An LSTM-based Maneuver Detection Algorithm from Satellites Pattern of Life." *2023 IEEE 10th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE, 2023.
6. Cipollone, R., Setya Ardi, N., Di Lizia, P. (2023). A Supervised Learning-Based Approach to Maneuver Detection Through TLE Data Mining. In: Ieracitano, C., Mammone, N., Di Clemente, M., Mahmud, M., Furfaro, R., Morabito, F.C. (eds) *The Use of Artificial Intelligence for Space Applications*. AII 2022. *Studies in Computational Intelligence*, vol 1088. Springer
7. DiBona, Phil, et al. "Machine learning for RSO maneuver classification and orbital pattern prediction." *2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. 2019.
8. Said Elsayed, M., Le-Khac, N. A., Dev, S., and Jurcut, A. D. (2020, November). Network anomaly detection using LSTM based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks* (pp. 37-45).
9. Forgy, Edward W. "Cluster analysis of multivariate data: efficiency versus interpretability of classifications." *biometrics* 21 (1965): 768-769.
10. Graves, Alex, and Alex Graves. "Long short-term memory." *Supervised sequence labelling with recurrent neural networks* (2012): 37-45.
11. Hinton, Geoffrey E., et al. "Parallel distributed processing: Explorations in the microstructure of cognition." (1986): 77-109.
12. Hoots, Felix R., Paul W. Schumacher Jr, and Robert A. Glover. "History of analytical orbit modeling in the US space surveillance system." *Journal of Guidance, Control, and Dynamics* 27.2 (2004): 174-185.
13. Kato, Ryo, et al. "Validity evaluation of anomaly detection using lstm autoencoder for maneuver detection." *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2023.
14. Kelecy, Tom, et al. "Satellite maneuver detection using Two-line Element (TLE) data." *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HA: Maui Economic Development Board (MEDB), 2007.
15. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
16. Lane, Ben, et al. "Using machine learning for advanced anomaly detection and classification." *Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS)*. 2016.
17. Lemmens, S., Krag, H. : Two-Line-Elements-Based Maneuver Detection Methods for Satellites in Low Earth Orbit. *Journal of Guidance, Control and Dynamics* 37(3) (2014)
18. Lindemann, B., Maschler, B., Sahlab, N., and Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, 131, 103498.
19. Mital, Rohit, et al. "A machine learning approach to modeling satellite behavior." *2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT)*. IEEE, 2019.
20. Patera, Russell P. "Space event detection method." *Journal of Spacecraft and Rockets* 45.3 (2008): 554-559.
21. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *the Journal of machine Learning research* 12 (2011): 2825-2830.
22. Perovich, Nicholas, Zachary Folcik, and Rafael Jaimes. "Satellite maneuver detection using machine learning and neural network methodsbehaviors." *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022.
23. Roberts, Thomas González. *Geosynchronous Satellite Maneuver Classification and Orbital Pattern Anomaly Detection via Supervised Machine Learning*. Diss. Massachusetts Institute of Technology, 2021.
24. Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." *IEEE transactions on Signal Processing* 45.11 (1997): 2673-2681.
25. Shabarekh, C., Kent-Bryant, J., Keselman, G., Mitidis, A. : *A Novel Method for Satellite Maneuver Prediction*. AMOS 2016.
26. Sidi, Marcel J. *Spacecraft dynamics and control: a practical engineering approach*. Vol. 7. Cambridge university press, 1997.
27. Vallado, David A., B. Bastida Virgili, and Tim Flohrer. "Improved SSA through orbit determination of two-line element sets." *ESA Space Debris Conference*. 2013.
28. Vallado, David, et al. "Revisiting spacetrack report 3." *AIAA/AAS astrodynamics specialist conference and exhibit*. 2006.
29. Zhou, C., and Paffenroth, R. C. (2017, August). Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 665-674).