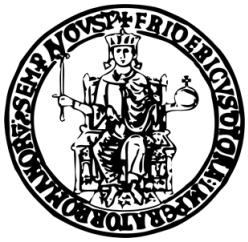
UNIVERSITY OF NAPLES FEDERICO II

DEPARTMENT OF INDUSTRIAL ENGINEERING

PH.D. COURSE IN INDUSTRIAL ENGINEERING

COORDINATOR PROF. MICHELE GRASSI

XXXVI CYCLE



PH.D. THESIS

EVENT, OBJECT AND MISSION CHARACTERIZATION FOR SPACE DOMAIN AWARENESS

By

Nicola Cimmino

Supervisors:

Prof. Giancarmine Fasano

Prof. Roberto Opromolla

This page intentionally left blank

Table of contents

Abstract

This PhD dissertation discusses the design, development and testing of innovative algorithms, methods and tools applicable in the context of Space Situational Awareness. Three main research topics have been addressed: fragmentation events characterization; use of Machine Learning approaches for ballistic coefficient estimation based on astrometric data; optimal manoeuvring strategy for responsive space applications.

In the context of Space Surveillance and Tracking (SST), operations such as detection, monitoring, and characterization of space objects are crucial for various functions, such as accurate orbit determination and collision risk assessment. In recent years, growing interest has been shown for space debris due to the lack of knowledge about their physical characteristics and thus the inaccurate determination of their orbits. To address this, a first research area of this PhD activity focused on improving the characterization of in-orbit fragmentation events, which represent a significant source of space debris. Specifically, starting from the implementation of the NASA's Standard Breakup Model, a primary contribution of the research activity was to develop an iterative approach that improves the estimation of the fragmented mass of two space objects colliding in orbit, leveraging the number of catalogued fragments after a certain time from the fragmentation event.

The second research path involved the development of Machine Learning (ML) approaches for estimating the ballistic coefficient (BC) of resident space objects in Low Earth Orbit, exploiting astrometric data. Various ML techniques were explored, particularly two types of neural network architectures: Feedforward Neural Networks and Recurrent Neural Networks. In both cases, the sensitivity of these architectures to the quantity and quality of data was analysed, especially robustness to measurement errors. Additionally, the effects of solar activity and atmospheric model uncertainty on BC estimates were evaluated. The ML approach was then compared with conventional characterization techniques.

Finally, in the complex and congested space environment, the ability of space assets to react to unforeseen situations is essential. In this context, an optimal multi-satellite manoeuvring strategy was proposed, to ensure the observation of areas of interest on the Earth's surface. Moreover, tools to support several Recognised Space Picture functions were developed, such as visibility analysis from ground-based sensors as well as GNSS coverage.

Keywords: space domain awareness; space situational awareness; space surveillance and tracking; breakup model; machine learning; neural networks; resident space object characterization; recognized space picture; responsive space

1. Introduction

This chapter provides an overview of the international context in which this PhD dissertation has been developed as well as of the state of the art of current literature regarding fragmentation event, resident space objects and space mission characterization. Moreover, the thesis structure is described.

1.1 Space Domain Awareness context

In recent years, there has been an ever-increasing number of Resident Space Objects (RSOs), namely, any human-made object orbiting around the Earth or residing in space for an extended time period [1], [2], [3]. This is due to advancements in miniaturised electronics, commercialization and standardization of satellite subsystem components (e.g., CubeSat standard), and increased accessibility to launch vehicles as secondary payloads, which have reduced the barriers to space. Furthermore, several commercial entities have either planned or initiated the launch of mega constellations (e.g., Startlink, OneWeb, Kuiper) to provide global surface imaging and space-based internet access. While this trend enables delivering worldwide services, it has also raised the probability of on-orbit collisions and satellite breakup. These hazards pose a threat to space-based assets, and can also jeopardise the safety of assets and people on the ground, especially in the case of re-entry of large uncontrolled objects. Currently, more than 34,000 catalogued objects are on orbit, whereas the estimated number of space debris larger than 1 cm is more than 1 million [4]. This leads to a congested and hazardous environment, posing long-term sustainability challenges.

These challenges underscore the increasing necessity and benefits of a Space Domain Awareness “to predict, avoid, […], recover from, and/or attribute cause to the loss and/or degradation of space capabilities and services.” [5]. To this aim, several Space Situational Awareness programs arose in the last years [6], aligning with space debris mitigation guidelines defined by the Committee on the Peaceful Uses of Outer

Space (COPUOS) [7]. The European Space Agency (ESA) delineates three key segments in SSA [8]:

- Space surveillance and tracking (SST), to monitor objects in Earth orbit including active and inactive satellites, spent rocket bodies and other debris fragments (either breakup-event related or mission-related);
- Space weather (SWE) to monitor conditions at the Sun's surface and in the solar wind, and in Earth's magnetosphere, ionosphere and thermosphere, that can affect space-borne and ground-based infrastructures or endanger human life or health;
- Near-Earth objects (NEO), to detect natural objects that can potentially impact Earth and cause damage.

It is evident that, within the SST segment, alongside with mitigation and remediation policies (e.g., satellites end-of-life disposal regulations and active debris removal) [9], [10], [11], [12], [13], [14], an imperative task is to detect, track and characterize the debris population, whose characteristics are usually unknown or known with very low accuracy, thus preventing precise orbit determination and propagation. A major source of space debris stems from breakup events, namely, on orbit collisions and explosions [15], [16], [17]. These events are hard to characterize since they depend on several parameters such as size, mass, shape of the object (e.g., cubic, hollow cylinders, plates), and materials (e.g., aluminium, titanium, composites). Additionally, the causes of explosions are also relevant: electrical malfunctions, such as short circuits, result in distinct explosion types compared to propellant autoignition. In the case of collisions, instead, the relative geometry and velocity between the colliding objects are critical factors.

Several efforts are underway to characterize fragmentation events: experimental analyses (such as on-ground hypervelocity tests) [18], physics-based analytical and/or numerical breakup modelling [19], and statistical approaches [20]. The main output of these analyses is the estimation of the physical parameters (e.g., cross-sectional area, mass, characteristic length) and of the kinetic energy (hence, of the velocity) of the generated fragments. These characteristics are crucial for modelling and propagating the resulting debris cloud, enabling the trajectory prediction and the assessment of

collision risks with operative satellites or other resident space objects. The NASA Standard Breakup Model [21] is a widely used satellite breakup model. However, several works have demonstrated that, the number and the characteristics of the simulated fragments are different to the ones generated by a real breakup event, in particular in the case of on-orbit collisions. The reason is that the SBM is purely energy-based and does not consider the relative attitude between the two colliding objects nor the point of collision. Therefore, a good estimation of the fragmented masses of the parents is critical to correctly set the input parameters of the SBM.

Within this PhD thesis, this aspect has been tackled proposing an iterative approach to estimate the fragmented mass of the parent objects in case of collision events. The iterative logic exploits the knowledge of Two-Line Elements of the fragments at some time after the event to adjust the input parameters of the breakup model, with the objective of obtaining the same number of real fragments, with a certain tolerance. The SBM has been interfaced with a backward numerical propagator to determine the catalogued objects' trajectory back to the fragmentation event epoch and separate them into two clusters corresponding to the target and projectile debris clouds. This allows performing the iterative approach on the two parent objects separately and, hence, estimating their fragmented masses. As a result, the breakup model outputs a set of fragments whose statistical distribution, in terms of number and size, is consistent with the catalogued ones.

Despite breakup models serve as valuable tools for understanding the dynamics of fragmentation events and assessing potential consequences in space debris scenarios, they face several limitations. These models often rely on simplifications and assumptions, which may not fully capture the complexity of the breakup process, introducing uncertainties in predictions. Accurate representation of material properties, scaling effects, impact conditions, and interactions among debris pieces poses challenges, and the inherent uncertainties in predicting outcomes make the models sensitive to the characteristics of each event. Besides space debris, characterizing operative satellites is likewise fundamental for various reasons. It enables precise tracking of satellite orbits, supporting collision avoidance manoeuvres, and enhances effective space traffic management (STM). This practice becomes

particularly significant for military purposes, where characterizing unfriendly space assets is paramount. Understanding the capabilities, orbits, and activities of adversary satellites is a critical component of military intelligence. This information allows military forces to assess potential threats, strategically plan defensive and offensive measures, and develop countermeasures to protect friendly satellites.

Current literature proposes several approaches to characterize RSOs, based on the type of available data (i.e., astrometric and/or photometric), the type of approach (i.e., physics-based and/or data-driven) and the physical parameters that are to be estimated (e.g., mass, size, cross section area, surface materials, drag coefficient, coefficient of reflectivity). Regarding the type of input data, astrometric information obtained from radars and telescopes observations provide information about an object's state (i.e., position and velocity). Conversely, photometric data (in the form of light curves, i.e., object brightness as seen from a telescope as a function of time) can be processed to provide information about the object's attitude evolution over time as well as its shape and size.

Two main types of approaches can be identified: physics-based and data-driven methods, each with its distinct advantages and limitations. Physics-based approaches [22], [23], [24], [25], [26], [27] rely on established theoretical models derived from fundamental physical principles. These models offer a deep understanding of the underlying physics governing space object behaviour, enabling accurate predictions and robust interpretations. However, the complexity of real-world scenarios can challenge the precision of these models, and uncertainties in environmental conditions or object properties may impact the accuracy of predictions. On the other hand, data-driven approaches [28], [29], [30], [31], [32], [33], [34], [35] harness the power of empirical data to characterize space objects. Machine learning algorithms aim at identifying patterns and relationships within wide datasets. Nevertheless, data-driven methods heavily depend on the quality and quantity of available data, and they might struggle with extrapolating beyond the observed data range.

This PhD thesis explores the application of Machine Learning (ML) approaches for recovering the ballistic coefficient (BC) from astrometric data, with a specific focus on Low Earth Orbit (LEO). Low Earth Orbit (LEO), in fact, is a central concern in

Space Situational Awareness (SSA) because of its high density of satellites, hosting critical assets for Earth observation and disaster response. Moreover, the rapid orbital decay in LEO, due to the atmospheric drag, highlights the need for precise tracking to predict satellite re-entries. In this region, the BC plays an important role, being the main parameter that affects the atmospheric drag acting on the RSOs. Two different architectures are explored for this application: feedforward and recurrent neural networks [36]. Feedforward Neural Networks (FNNs) are usually more simple and computationally efficient than Recurrent Neural Networks (RNNs). However, they lack memory of past inputs, limiting their suitability for tasks involving sequential or time-dependent data. On the other hand, RNNs excel in capturing temporal dependencies and handling sequential data. However, they can face challenges such as vanishing or exploding gradient problems during training and can be computationally expensive. The use of a high-fidelity simulation environment for training, validation, and testing, along with real-world test cases, demonstrates the applicability of the proposed ML techniques, making it a valuable tool for SSA applications in operational settings.

From what has been described so far, it is clear that, in the context of SDA, a transversal fundamental service is the Recognized Space Picture (RSP) which, among other functions, provides detection, tracking (i.e., space surveillance) and identification (i.e., space reconnaissance) of artificial space objects as well as ground and space threats assessment on space related activities, services and operations (i.e., threats assessment) [37]. RSP relies on a wide net of ground-based and spaceborne sensors.

Within this PhD thesis, different tools to support these functions have been developed. First, a tool to evaluate whether a satellite overflies a designated Area of Interest (AoI) is developed, assuming the availability of a specific set of ground-based sensors. Second, it has been developed a tool to assess the capability of a given sensor-equipped satellite to observe a predetermined AoI. Finally, the thesis presents a tool for computing the Dilution of Precision (DOP) of a set of GNSS satellites on a predefined AoI.

In such a congested and complex orbital environment, an important capability of space assets is their prompt response to unforeseen situations [38], [39]. Scenarios such as unexpected close approaches between satellites, potential collisions, or sudden space weather events, which can impact the operational capabilities of satellites, may require swift responses to ensure the safety and operational integrity of space assets. Furthermore, unforeseen national security threats, satellite anomalies, or shifts in operational requirements (such as those prompted by local emergencies) may demand the ability for agile manoeuvring.

In this PhD thesis, an important contribution has been provided to enhance satellites responsiveness. In particular, an optimized multi-satellite manoeuvring strategy for responsive Earth observation has been proposed and its applicability to real scenarios has been demonstrated.

It is worth outlining that part of the research activity has been realized in collaboration with the Engineering Group for Aerospace Gruppo (GIAS) of the Italian Air Force, and during a period of six months spent at Imperial College London (UK).

1.2 Thesis organization

The thesis is organised as follows:

- Chapter 2 addresses the problem of fragmentation events characterization. After providing an overview on the state-of-the-art regarding breakup models, it describes the Standard Breakup Model. Then, it explains the iterative approach implemented to estimate the fragmented mass of the parent objects. The iterative approach is then demonstrated using simulated and real scenarios.

- Chapter 3 tackles the problem of Resident Space Objects characterization. It provides an overview of current literature approaches to estimate several physical parameters, both physics-based and data-driven. Then, it describes the proposed

Machine Learning-based methodologies, exploring different Neural Networks architectures. It presents the sensitivity analyses carried out and the applicability of the proposed methods to real test cases. Moreover, a comparison between the ML approach and a semi-analytical methos is presented.

- Chapter 4 presents the tools developed in the RSP context and the results obtained with numerical simulations. Then, in the context of responsive space, the optimal manoeuvring strategy is described as well as its application to simulated and real scenarios.

- Chapter 5 contains the conclusions, to summarize the results and anticipate further developments of the research activity.

2. Fragmentation event characterization

This chapter addresses the problem of fragmentation events characterization. After providing an overview on the state-of-the-art regarding breakup models, it describes the Standard Breakup Model. Then, it explains the iterative approach implemented to estimate the fragmented mass of the parent objects. The iterative approach is then demonstrated using simulated and real scenarios.

2.1 Literature review

As previously mentioned in Section 1.1, Space debris is a threat to the safe operation of satellites and the long-term sustainability of space activities. The generation of space debris via collisions and explosions in orbit could lead to an exponential increase in the number of artificial objects in space in a chain reaction known as Kessler Syndrome [4]. Nowadays, among the more than 8700 objects larger than 10 cm in Earth orbits, only about 6% are operational satellites, whereas the remainder is space debris. In addition, there are many smaller objects that are not routinely tracked, with estimates for the number of objects larger than 1 cm ranging from 100,000 to 200,000 [40] and an estimated 128 million particles between 1 mm and 1 cm [41]. Fragments of 1 cm size are particularly critical since they are very hard to detect, even in lower orbits, but can be mission-ending for operating satellites due to their high speed and, consequently, associated impact energy.

The dominating source of space debris by number of objects are breakup events, mainly due to on-orbit explosions and collisions. There are several reasons for breakup events, labelled by ESA Database and Information System Characterising Objects in Space (DISCOS) as follows [42]: i) Propulsion: stored energy for non-passivated propulsion-related subsystems might lead to an explosion, for example due to thermal stress; ii) Upper stages explosions due to residual propellants; iii) Ullage motors; iv) Electrical: overcharging and subsequent explosion of batteries; v) Aerodynamics:

breakup most often caused by an overpressure due to atmospheric drag; vi) Collision between two objects called *parents*, usually referred to as *target* (i.e., the primary or larger object) and *projectile* (i.e., the secondary or smaller object); vii) Deliberate: intentional breakup events; viii) Anti-satellite tests (ASAT); ix) Anomalous: defined as "the unplanned separation, usually at low velocity, of one or more detectable objects from a satellite that remains essentially intact".

Some of the events result in only a few objects, which might even quickly decay due to the atmospheric drag. In all the other cases, a large quantity of fragments is generated, thus significantly contributing to the increase in the collision risk that satellites are facing during their operational life [42]. While implementing debris mitigation policies, space agencies and academia continuously develop models and software to analyse the space environment, estimating the collision probability between catalogued space objects and predicting future growth of debris population [40]-[42], [43], [44], [45], [46], [47], [48], [49].

A fundamental component of any orbital debris environment evolution model is the satellite breakup model. The worldwide reference is the NASA Standard Breakup Model (SBM) implemented both in EVOLVE 4.0 [21] and later in LEGEND [47]. It provides three fundamental outputs: fragment size, area-to-mass ratio (*AMR*), and relative velocity (ΔV) distributions [50]. Such distributions have been obtained by a broad variety of data coming from deliberate hypervelocity collisions in LEO, ground-based impacts, and upper stage explosion tests, as well as from an extensive compilation of historical orbital data (i.e., two-line element sets) for debris generated by explosions and collision [21].

Due to the semi-empirical and statistical nature of the SBM, there are two major research paths concerning breakup models. On the one hand, there is the need to continuously update the list of catalogued space objects to have better data fitting. For this reason, a major effort, made by different players in the space industry, has been to track and image space objects down to 1 cm at altitudes up to 1000 km, and down to 2 mm at low LEO altitudes [40] thanks to the development of high performance ground-based RADAR systems (e.g., SpaceFence, Haystack, TIRA, LeoLabs) [51], [52], [53]. Moreover, other works aim to improve collision modelling via tests on new

generation satellites, as a clear trend can be seen towards smaller and lower mass spacecraft [54]. Simulations on Cosmos 2251– Iridium 33 collision and Fengyun-1C ASAT test have demonstrated that while Cosmos 2251 fragments are well-described by the NASA SBM, noticeable discrepancies exist between the model predictions and the observation data for the Iridium 33 satellite and Fengyun-1C ASAT. The justification is that Cosmos 2251 was an older satellite while Iridium 33 and the Fengyun-1C weather satellites were relatively modern [50], [55], [56], [57]. Hence, NASA DebriSat project aims to characterize fragments generated by hypervelocity collisions involving a modern satellite in low Earth orbit (LEO). There are three phases to this project: the design and fabrication of DebriSat - an engineering model representing a modern, 60 cm/50 kg class LEO satellite; conduction of a laboratory-based hypervelocity impact to catastrophically break up the satellite; and characterization of the properties of breakup fragments down to 2 mm in size [50], [55], [56], [57]. Similarly, other experiments have been carried out to characterize micro-satellite impacts (20 cm/1.5 kg), both at hyper and low velocity, varying impact directions and investigating fragments from multi-layer insulation and solar cells [58], [59]. Such experiments have demonstrated that, although the SBM well represents the size distribution, mass and *AMR* distributions are greatly influenced by the materials adopted [58], [59].

Other software programs, like IMPACT [57], [60] introduce other inputs in the breakup model to better characterize the physical properties distribution of the fragments: the type of object (i.e., hollow or compact), the material density or, in case of explosion, the amount of energy driving the explosion (e.g., chemical energy, pressure, electrical energy). Moreover, while in the SBM the standard condition for a complete fragmentation is that the kinetic energy of the projectile relative to the target divided by the target mass is greater than 40J/g, IMPACT introduces a further value of 10 J/g to smooth the transition from a typical hypervelocity collision with large numbers of small fragments to lower energy collisions where most of the target object is broken into a few larger fragments [20].

The second research area regards the estimation of the input data of the SBM to model real fragmentation events. In the case of collisions, the input data are the mass

of the two colliding objects, M_1 and M_2, and the impact velocity v_{imp} (i.e., the norm of the relative velocity). In the case of explosions, instead, an important role is played by the scale factor of the SBM, assuming a default value of 1 for upper stages with mass between 600 kg and 1000 kg [21]. Several works [42], [61], [62], [63], however, have demonstrated that, giving such inputs, the number and the characteristics of the simulated fragments are different to the ones generated by a real breakup event. In particular, Anz-Meador and Matney [61] and Braun et al. [42] have demonstrated that several debris clouds derived from explosions of nominal upper stages are represented by values of the scale factor significantly different from 1. Similarly, Pardini and Anselmo [63] show that, out of four on-orbit accidental collisions, all having an energy-to-mass ratio greater than 40 J/g, only one has generated a number of fragments similar to the one predicted by the SBM. The reason is that the breakup model is purely energy-based and does not consider the relative attitude between the two colliding objects nor the point of collision. Indeed, the basic implementation of the SBM assumes that collisions will always involve the entire body of each parent. This does not account for spacecraft consisting of multiple connected structures, for instance the presence of appendages such as gravity gradient booms or deployed solar panels. In this respect, the effect of a collision is clearly different if the projectile hits the main body of a satellite or only a part of it. In the latter case, the fragmented mass of the two objects will be probably lower and the breakup model would overestimate the number of the debris produced. Similarly, for explosions the uncertainty regards the energy driving the breakup and, hence, the actual exploded mass. It is clear, then, that a good estimation of the fragmented masses of the parents is critical to correctly set the input parameters of the SBM.

For the estimation of the scale factor, a simple procedure can be adopted as in [64]: assuming that a certain number of fragments (N^*) over a minimum trackable size has been catalogued, the scale factor can be derived explicitly from the equation of the cumulative distribution of the number of fragments with respect to the characteristic length. Another approach proposed by [42] can be applied to both explosion and collision events. It consists of a dynamic scaling approach using a reported number of observed fragments (or number of catalogued fragments) multiplied by an altitude dependent catalogue incompleteness factor (so-called Henize factor) to scale the power

law distribution. This procedure is used by [42] to estimate the scale factor, for explosions, and the total fragmented mass of the parents, for collisions. The Henize factor however presents a strong discontinuity for L_c equal to 6 cm.

In this framework, this PhD thesis proposes an alternative way to estimate the tuning parameters for the SBM, focusing on the fragmented mass of the parents in case of collisions. The method requires TLE sets from the event to be analysed. It assumes that enough time from the fragmentation event passed to have all the fragments over a certain size threshold catalogued. Moreover, decayed fragments are estimated. Hence, the breakup model is iterated until the number of simulated fragments over a predefined threshold of the characteristic length is equal to the number of real fragments, within a certain tolerance. The iteration is performed on the masses of the two parents separately after identifying the debris clouds derived from the target and the projectile. Finally, the fragmented masses of the parents are estimated using the bisection method.

An important assumption in this paper is that the parents and their orbits are known. However, a correct parent identification is not always guaranteed, thus leading to erroneously assigned objects. Several works proposed methods to identify the epoch of a fragmentation as well as the known objects involved. In particular, Frey et al. [65] propose a method to detect past fragmentations exploiting the convergence of mean Keplerian elements when propagating backwards in time. Tetrault et al. [66] developed a tool that determines the objects involved in the fragmentation event using back propagation and particle distribution techniques. Dimare et al. [67], instead, define a similarity function between the orbital elements of the objects under examination to establish a metric for the identification of the fragmentation. The parent objects are found by locating the minimum of the similarity function among the various objects. Muciaccia et al. [68] use backward propagation, pruning, and clustering algorithms to identify possible orbital intersection windows that make close encounters between objects possible. To consider erroneous assignment of the fragments to the parents, an analysis of the method performance under this circumstance is also analysed.

2.2 Review of the Standard Breakup Model

The SBM models the fragment size, *AMR*, and ΔV distributions characterizing the debris generated by a fragmentation event [55]. In this section a synthetic description of the SBM, as implemented in EVOLVE 4.0, is reported. For more details regarding the derivation of the equations and the coefficients used in the statistical distributions, interested readers can refer to [21].

The SBM identifies two types of fragmentation events: collisions and explosions. In case of collisions, the input data are the masses of the parents (i.e., the two colliding space objects) and the impact velocity. According to the model, collisions can be either *catastrophic*, if the entire masses of the two objects are fully fragmented, or *non-catastrophic* in case of fragmentation of the projectile and cratering of the target. The distinction between *catastrophic* and *non-catastrophic* collisions can be done by evaluating the relative kinetic energy of the projectile divided by the mass of the target, as shown below,

$$E_p = \frac{1}{2}\frac{m_{proj}[kg]\left(v_{imp}[m/s]\right)^2}{m_{targ}[g]} \tag{2.1}$$

where m_{targ} and m_{proj} are the masses of the target and projectile, respectively, while v_{imp} is the impact velocity. Specifically, the collision is defined as *catastrophic* if the value of E_p is greater than 40 J/g. In the case of *non-catastrophic* collisions, i.e., E_p is less than 40 J/g, the fragmented mass of the parents is calculated as follows [21],

$$m_1 = m_{proj}\left(\frac{v_{imp}[km/s]}{1[km/s]}\right) \tag{2.2}$$

$$m_2 = m_{targ}\left(\frac{v_{imp}[km/s]}{1[km/s]}\right) \tag{2.3}$$

where m_1 and m_2 are the fragmented masses of the target and the projectile, respectively.

In the case of explosions, instead, the input data are the mass of the parent and a scale factor that has a unitary value for upper stages with a mass between 600 kg and 1000 kg.

In both cases, the cumulative distribution of the number of fragments with respect to the characteristic length L_c, is calculated using the following equations,

$$N = 0.1 \left(\frac{m_{frag}[kg]}{1[kg]} \right)^{0.75} \left(\frac{L_c[m]}{1[m]} \right)^{-1.71} \tag{2.4}$$

$$N = 6\, s \left(\frac{L_c[m]}{1[m]} \right)^{-1.6} \tag{2.5}$$

where N is the number of fragments, m_{frag} is the fragmented mass, and s is the scale factor. Then, the distribution of the number of fragments with respect to the *AMR*, for each characteristic length interval, is computed. For each pair of L_c – *AMR* intervals, values of characteristic length and area/mass ratio are extracted and then assigned to each fragment. It is assumed that within each interval there is a uniform probability distribution. The value of the area (A) for each fragment is computed as a function of its length L_c, using the following equation, valid both for collisions and explosions.

$$A = 0.556945\, L_c^{2.0047077} \tag{2.6}$$

The mass value M of each fragment can then be estimated as follows.

$$M = A \left(\frac{M}{A} \right) \tag{2.7}$$

Finally, the distribution of the speed variations (ΔV) due to the event is computed and a ΔV value is assigned to each fragment by random extraction.

2.3 Breakup Model Sensitivity Analysis and Tuning Approach

The input data of the collision model are the masses of the parents and the impact velocity. The impact velocity (v_{imp}), normally computed from the state vector of the parents generally affects the fragmented mass during a collision. However, considering that the typical uncertainty in the knowledge of the impact velocity is in the order of m/s [69], this effect is very limited. Moreover, as long as the number of catalogued fragments is known, the impact velocity does not influence the iterative estimation of the fragmented mass but its computational cost. In fact, (v_{imp}) is used only for the computation of the specific kinetic energy and, hence, for the definition of the type of collision (i.e., catastrophic or non-catastrophic). Such definition determines the initial values with which the iterative algorithm starts. Therefore, the impact velocity does not affect the final value of the estimated fragmented masses but only the number of iterations needed to reach convergence. In particular, if the collision is catastrophic, the impact velocity has no effect on the number of iterations, since the initial value of the iterative approach would always be equal to the total mass of the parents, regardless of the value of v_{imp}. In case of a non-catastrophic collision, the impact velocity instead enters directly in Eq. (2.2) and (2.3) to compute the estimated fragmented mass according to the SBM. This could lead to an increase of number of iterations, e.g., from 2 to 5. Based on these considerations, the masses of the parents are the only parameters to tune the model. The tuning of the masses can be done based on the accurate information provided by the catalogued fragments: namely, their number and orbital parameters. The orbital parameters available for the catalogued fragments can provide useful information on the ΔV distribution. However, such distribution seems to be rather insensitive to the input masses. To support this statement a sensitivity analysis has been carried out. For a fixed value of the projectile mass, set equal to 100 kg, the SBM has been run for three different values of the target mass M_{targ} (i.e., 100 kg, 500 kg and 1000 kg), and three different values of the impact velocity v_{imp} (i.e., 1 km/s, 5 km/s and 10 km/s). For each couple (M_{targ}, v_{imp}), 100 runs have been executed. For each run the mean μ and the standard deviation σ of the

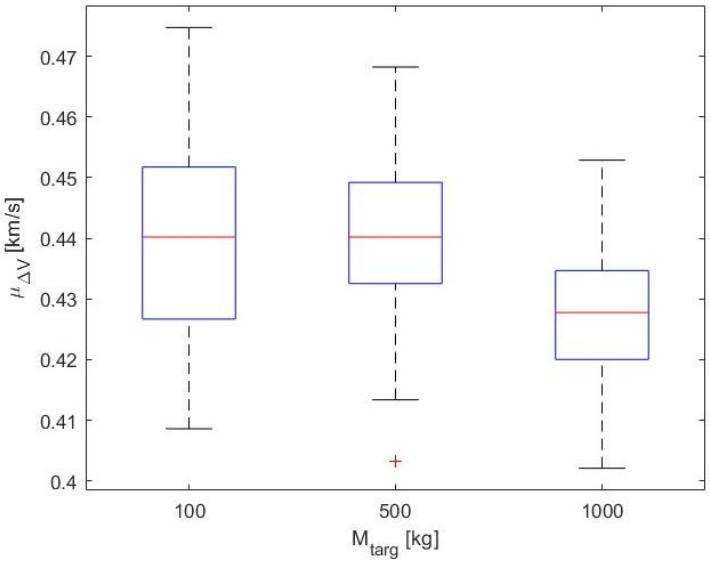
ΔV distribution have been computed for all the fragments generated by the breakup (hereinafter called *population*) and for the fragments with a characteristic length greater than 10 cm (hereinafter called *sample*). Therefore, for each couple (M_{targ}, v_{imp}), a set of 100 means ($\mu_{\Delta V}$) and standard deviations ($\sigma_{\Delta V}$) is obtained. Figure 2.1 – Figure 2.3 show the boxplots of $\mu_{\Delta V}$ and $\sigma_{\Delta V}$ in the different cases for the sample fragments. On each box, the central mark (i.e., the red line) indicates the median of the statistics parameter (i.e., $\mu_{\Delta V}$ and $\sigma_{\Delta V}$), the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The difference between the 75th and 25th percentiles (i.e., the height of the box) is called interquartile range (IQR). The whiskers extend to a value equal to 1.5 IQR above the 75th percentile and 1.5 IQR below the 25th percentile. Outliers are the values beyond the whiskers and are plotted individually using the '+' symbol. By looking at the mean and standard deviation distributions, it is clear that, for a fixed impact velocity, the ΔV distributions are statistically similar regardless of the target mass. In fact, applying a two-sample t-test to the ΔV distributions obtained with the same impact velocity and different target mass, the null hypothesis (i.e., the sample data are from populations whose expected means are the same), is accepted with 95% of confidence in all the cases. Moreover, the percentage variation of the median of the $\mu_{\Delta V}$ and $\sigma_{\Delta V}$ distributions has been computed for target mass equal to 500 kg and 1000 kg with respect to the case in which the target mass is equal to 100 kg. The results of this analysis are reported in Table 2.1 considering different values of the impact velocity. It can be noticed that the percentage variation is less than 5% for the median of $\mu_{\Delta V}$ and less than 10% for the median of $\sigma_{\Delta V}$.

Table 2.1 - Absolute percentage variation of $\mu_{\Delta V}$ and $\sigma_{\Delta V}$ distributions for target mass equal to 500 kg and 1000 kg with respect to target mass equal to 100 kg.

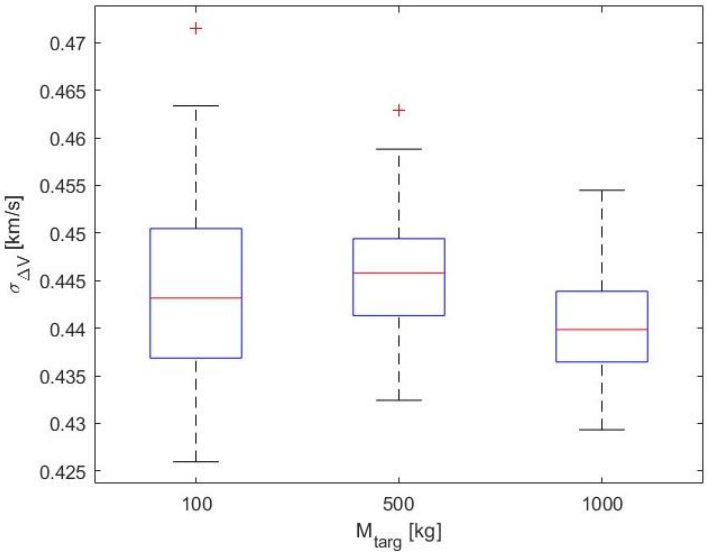
	Absolute percentage variation of $\mu_{\Delta V}$ [%]		Absolute percentage variation of $\sigma_{\Delta V}$ [%]	
M_p / v_{imp}	500 kg	1000 kg	500 kg	1000 kg
1 km/s	$0.3\ 10^{-2}$	2.8	0.6	0.7
5 km/s	1.4	0.9	6.6	9.2
10 km/s	2.1	4.9	2.5	2.0

To further support the above statement, Figure 2.4 plots the median of the means of ΔV distributions against the impact velocity for the different values of the target mass. It can be noticed how there is not a clear trend between the median of the mean of ΔV and the parent's mass. This analysis shows that, for a known impact velocity, the velocity distribution of the fragments is not significantly affected by the parent masses and thus cannot be used to estimate the fragmented mass.

Therefore, as in previous works [42], [64] the parameters estimation is based on the number of catalogued objects. This approach, however, presents three main challenges. First, the limitations of ground sensors introduce an observability gap. In fact, depending on the altitude, only fragments greater than a certain threshold, L_{min}, can be detected and tracked (e.g., L_{min} of order of 10 cm size at LEO and 1m at GEO). Second, after a short time from the fragmentation event (FEV), the cloud of debris is not dispersed enough, and it is hard to correctly catalogue the different fragments of the cloud. Third, especially for high energy collisions or for breakup events at lower orbits, a significant number of fragments decay due to the atmospheric drag. Thus, the epoch at which the fragments TLEs are known is important.

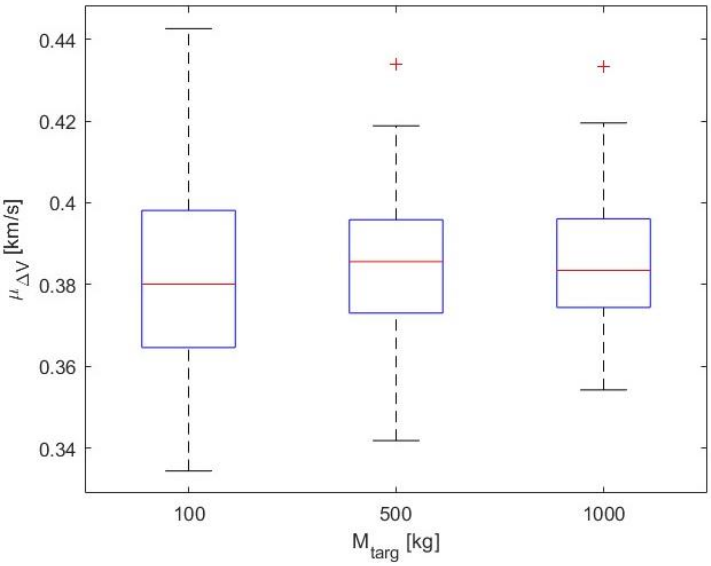


(a)

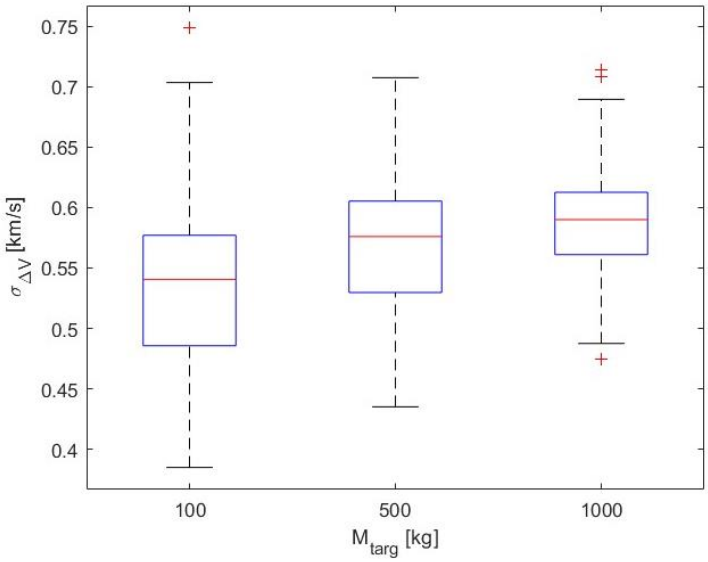


(b)

Figure 2.1 - Mean (a) and standard deviation (b) of the velocity distribution of the sample fragments for different values of the target mass for an impact velocity of 1 km/s. The mass of the projectile is assumed to be 100 kg.

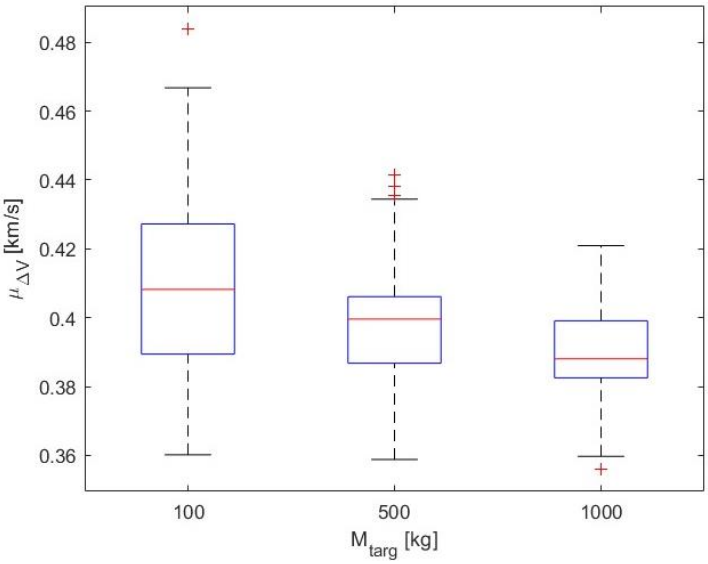


(a)

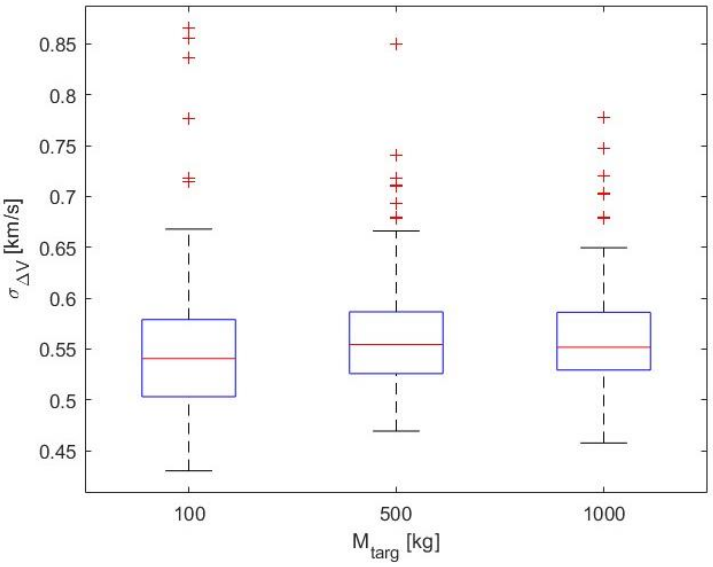


(b)

Figure 2.2 - Mean (a) and standard deviation (b) of the velocity distribution of the sample fragments for different values of the target mass for an impact velocity of 5 km/s. The mass of the projectile is assumed to be 100 kg.

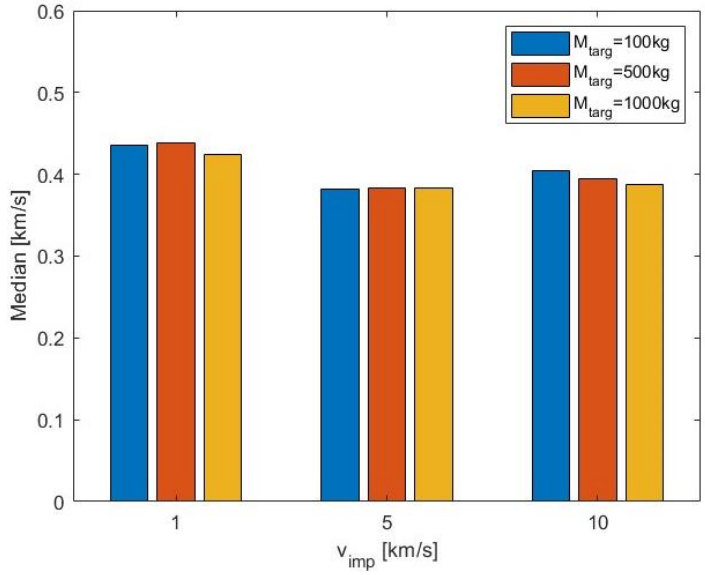


(a)

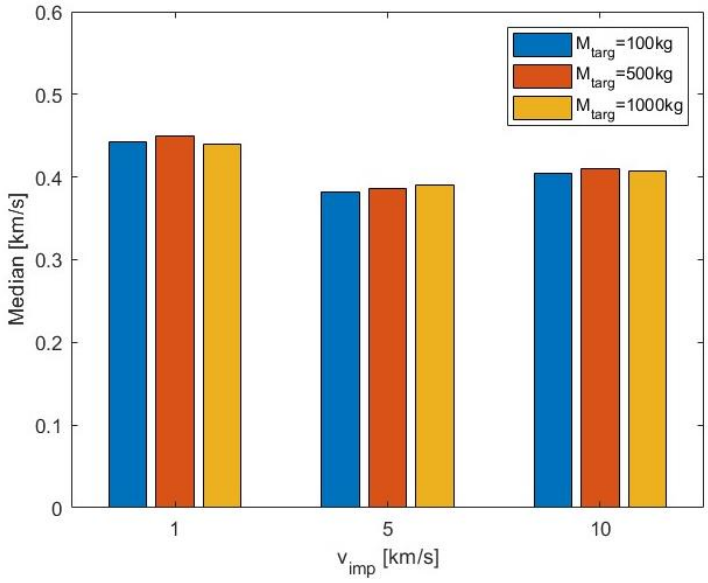


(b)

Figure 2.3 - Mean (a) and standard deviation (b) of the velocity distribution of the sample fragments for different values of the target mass for an impact velocity of 10 km/s. The mass of the projectile is assumed to be 100 kg.



(a)



(b)

Figure 2.4 - Median of the target (a) and projectile (b) fragments sample velocities for different values of the impact velocity and the target mass. The mass of the projectile is assumed to be 100 kg.

To better understand how to tune the model and to support the criteria adopted to filter the fragments generated by the SBM, a brief description of how the breakup model is implemented is provided. A scheme of the overall architecture is reported in Figure 2.5.

The developed model takes as inputs:

1. The type of input file to load (i.e., ephemerides or TLEs). In both cases, the software extracts the state vector of the parent(s), $\underline{\boldsymbol{x}}$, either directly, in the case of ephemerides, or converting the orbital parameters to position and velocity, in case of TLEs. Moreover, the impact velocity is computed as the norm of the relative velocity difference between the two colliding objects.

2. Mass and size of the parents

After the kinetic energy estimation and the classification of the type of collision, the distributions of the fragments' characteristics are estimated, as described in the

previous section. Such distributions can be limited within user-defined intervals based on literature [21]. In particular, at each step of the area/mass extraction (and hence of each fragment mass assignment), a check on the mass conservation principle is carried out. Specifically, fragments are extracted according to the normal procedure as described in Section 2.2, until the difference between the sum of their masses (M_{tot}) and the mass of the parent is lower than a tolerance threshold (i.e., 5%). If the extraction procedure leads to a value of M_{tot} greater than the mass of the parent by more than 5%, the fragments with minimum *AMR* probability, in the current L_c bin, are removed. Then the ΔV distribution is estimated. The SBM only provides the distribution of the ΔV module for the fragments generated by the breakup event. To analyse the fragments' evolution, a direction must be assigned to each ΔV. This task must be done assuring that the conservation of momentum is met. Hence, an icosahedral grid has been created whose nodes correspond to directions in a three-dimensional space [62]. It is assumed that each node has the same probability to be associated to each fragment. The ΔV, multiplied by the direction cosines corresponding to each direction, gives the velocity components of each fragment. Each fragment is therefore characterized by length, mass, area, and area/mass ratio, position at the fragmentation event (FEV), coinciding with the position of the parents, speed at FEV, computed as the sum between the speed of the parent and the ΔV. The obtained set of fragments is then filtered by removing objects characterized by:

- length smaller than a threshold value (L_{thr}) defined by the user,
- negative semi-major axis (SMA_{thr}), corresponding to a hyperbolic orbit, not of interest for the analyses of this paper,
- altitude at perigee lower than 120 km (h_{thr}), corresponding to a fragment that will re-enter before being catalogued.

The previous filtering is applied after assuring the conservation of momentum is achieved.

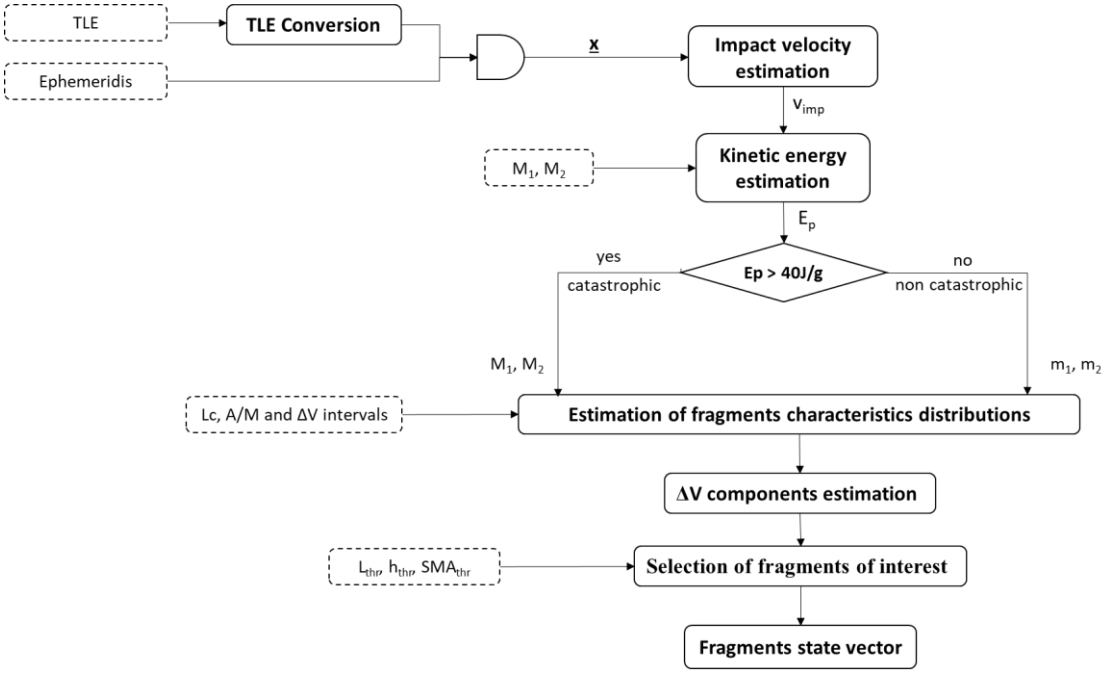


Figure 2.5 - Scheme of the SMB implementation. Each fragment is characterised by physical characteristics (i.e., size, mass, *AMR*) and a state vector (i.e., position and velocity).

2.3.1 The iterative approach

An iterative logic based on the TLEs of catalogued fragments relevant to a fragmentation event is adopted to determine the input parameters of the breakup model and obtain the same number of real fragments, within a certain tolerance. For the scope of this work, it is assumed that, at the altitude of interest, all the fragments greater than a pre-defined threshold have been catalogued. The logic consists of five main steps (Figure 2.6):

1. The TLEs of the fragments are loaded and the orbital parameters are extracted. Let N_{cat} be the number of catalogued fragments.

2. A set of simulated fragments is generated from the breakup model using as default values the mass of the parents and their relative velocity. Let N_{SBM,L_c} be the number of simulated fragments with a length greater than the pre-defined size threshold.

3. The catalogued fragments are backward propagated until the FEV. This helps to distinguish the two clusters of fragments belonging respectively to the target and the projectile, out of a TLE file containing mixed fragments generated by the parents. The general principle is to compute the norm of the relative velocity between fragments and parents at FEV, and to assign each fragment to the parent corresponding to the minimum norm. In the case of orbital planes rotated by at least a few degrees with respect to each other, which will typically characterize collision geometries for very low eccentricity orbits, the concept has a direct analytical implementation based on inclinations and right ascension of the ascending node. In fact, it is well-known that a plane change requires a large value of the out-of-plane component of ΔV. This is confirmed also from historic breakups, registering a maximum plane change of 3 degrees for the fragments with respect to their parent's original orbit [60]. Therefore, it can be assumed that, even with high energy collisions, the breakup fragments will only be dispersed a small amount from the parents' orbital planes. Using the law of cosines for spherical triangles, the rotation of each fragment's orbital plane with respect to target and projectile ones is computed:

$$\theta_{targ} = acos\big(cos\big(i_{frag}\big)\, cos\big(i_{targ}\big) + sin\big(i_{frag}\big)\, sin\big(i_{targ}\big)\, cos\big(\Omega_{targ} - \Omega_{frag}\big)\big) \tag{2.8}$$

$$\theta_{proj} = acos\big(cos\big(i_{frag}\big)\, cos\big(i_{proj}\big) + sin\big(i_{frag}\big)\, sin\big(i_{proj}\big)\, cos\big(\Omega_{proj} - \Omega_{frag}\big)\big) \tag{2.9}$$

where: θ_{targ} and θ_{proj} are the angles between the fragment orbit plane and the orbit plane of target and projectile, respectively, i_{frag}, i_{targ}, and i_{proj} are the inclination of the fragment, target and projectile orbit planes, Ω_{frag}, Ω_{targ} and Ω_{proj} are the right ascension of the ascending node of the fragment, target and projectile orbit planes. If θ_{targ} is smaller (larger) than θ_{proj} the fragment is assigned to the target (projectile).

This procedure, which assumes that the colliding objects have orbital planes rotated of at least a few degrees with respect to each other, allows splitting in two

parts the iterative process and carrying out two separate analyses. It must be highlighted that, especially when a short time since the FEV has passed, mass and cross section area of the fragments, necessary for the backward propagation, might be unknown. Therefore, a default value of *AMR* is assumed. However, the uncertainty on these values should not affect the out-of-plane dynamics and, hence, the assignment of the fragments to the two debris clouds.

4. Estimation of decayed fragments: due to a negative ΔV or to the atmospheric drag, some fragments may decay in the time frame, $\Delta t_{FEV,TLE}$, between the FEV epoch and the last available TLE epoch. The number of decayed fragments, N_{dec}, can be negligible in the iterative process when many catalogued fragments, N_{cat}, are available. However, when N_{cat}, is small, the number of decayed objects might represent a significant percentage thus affecting the estimation of the fragmented mass. To consider this aspect in the iterative process, N_{dec} must be estimated. This is done by counting all the N_{SBM,L_c} fragments having an altitude at perigee below 150 km. It is indeed reasonable to consider that the fragments with this characteristic decayed in the time frame $\Delta t_{FEV,TLE}$.

5. Iterative algorithm: the iterative algorithm operates until the following relationship is satisfied,

$$N_{cat} - tol \leq N_{Lc,nodec} \leq N_{cat} + tol \tag{2.10}$$

where tol is a tolerance margin, $N_{Lc,nodec}$ is the number of not decayed fragments larger than the threshold (i.e., $N_{SBM,L_c} - N_{dec}$). The tolerance value is set equal to 0.1 N_{cat} for a mass of the parents greater than 100 kg, 0.2 N_{cat} for a mass in the range 50 kg-100 kg, and 0.3 N_{cat} for a mass of less than 50 kg. Figure 2.6 shows a scheme of the iterative logic implemented.

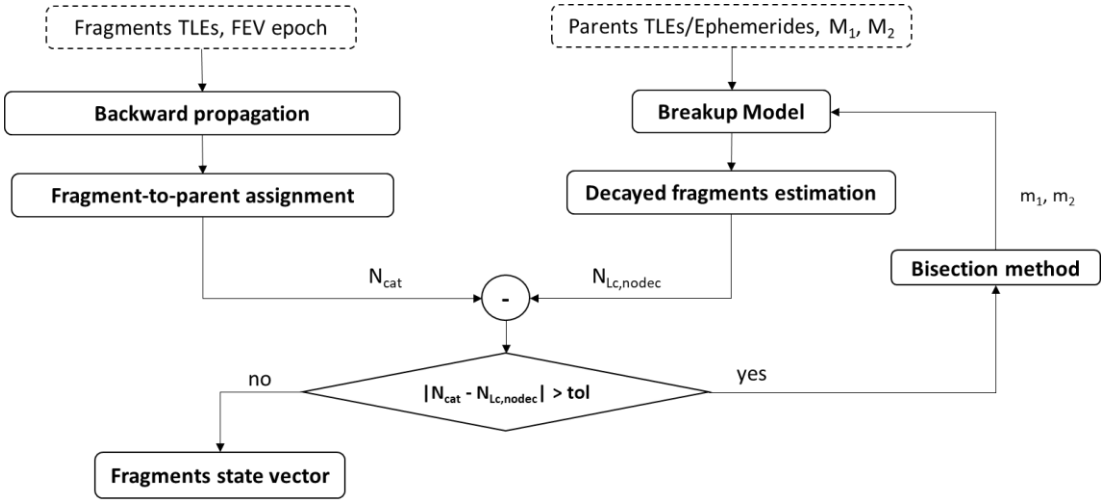


Figure 2.6 - Scheme of the iterative logic.

When equation (2.10) is not satisfied, the bisection method is applied to estimate the tuning parameters. As stated in the previous section, the tuning parameters are the fragmented masses of the two parents. The bisection method is applied to the number of target and projectile fragments separately.

The inputs to the algorithm are the number of catalogued fragments from the target and projectile clouds ($N_{cat,targ}$ and $N_{cat,proj}$), the number of simulated fragments previously generated, with altitude of perigee greater than 150 km at FEV ($N_{Lc,nodec}$) and the mass $M_{SBM,i}$, with i = 1, 2 (1 for the target and 2 for the projectile). $M_{SBM,i}$ is equal to the total parent mass (for catastrophic event) or the estimated fragmented mass by the SBM, according to equation (2.2-2.3). If the number of simulated fragments is equal (within a certain percentage) to the number of real fragments, the algorithm is not executed, and the number of simulated fragments is the one generated by the SBM. Otherwise, the algorithm is executed. Let M_i be the initial mass of the i-th parent (i.e., i=1 for the target and i=2 for the projectile); $m_{i,j}$ be the estimated fragmented mass for the i-th parent at the j-th iteration; $[a_j \ b_j]$ be the interval in which the bisection method is applied. For j = 1:

$$a_j = 0, b_j = M_i \tag{2.11}$$

then, if $N_{Lc,nodec} > N_{cat} + tol$:

$$m_{i,j} = \frac{m_{i,0}}{2} \tag{2.12}$$

both for catastrophic and non-catastrophic collision, while if $N_{Lc,nodec} < N_{cat} - tol$:

$$m_{i,j} = M_i \tag{2.13}$$

for catastrophic collision, and:

$$m_{i,j} = \frac{m_{i,0} + b_j}{2} \tag{2.14}$$

for non-catastrophic collision. For $j \geq 2$, both for catastrophic and non-catastrophic collisions:

$$m_{i,j} = \frac{a_j + b_j}{2} \tag{2.15}$$

where, if $N_{Lc,nodec} > N_{cat} + tol$:

$$\begin{bmatrix} a_j & b_j \end{bmatrix} = \begin{bmatrix} a_{j-1} & m_{i,j-1} \end{bmatrix} \tag{2.16}$$

while, if $N_{Lc,nodec} < N_{cat} - tol$:

$$\begin{bmatrix} a_j & b_j \end{bmatrix} = \begin{bmatrix} m_{i,j-1} & b_{j-1} \end{bmatrix} \tag{2.17}$$

Equations (2.15) to (2.17) are valid both for catastrophic and non-catastrophic collisions. In all the previous cases, the bisection method is lower bounded by zero mass and upper bounded by the initial parent's mass.

2.4 Performance assessment of the iterative approach

To demonstrate effectiveness of the proposed iterative approach, two simulated and one real collision events are considered as test cases.

2.4.1 Simulated and real collision events

A collision is hypothesized assuming the masses of the parent objects, i.e., M_1 and M_2, the impact velocity and the corresponding fragmented masses (i.e., m_1 and m_2). By running the SBM using m_1 and m_2 as input masses, the reference values for the number of generated fragments, i.e., N_1 and N_2, can be obtained.

The goal of the proposed iterative approach is to estimate m_1 and m_2 assuming that all the observable fragments, i.e., those with characteristic length larger than 10 cm, are catalogued. To this aim, the iterative approach is applied using the total masses of the parents, i.e., M_1 and M_2, as initial guess. The SBM is then iterated until the convergence condition given by Eq. 2.10 is met. At convergence, the number of generated fragments is indicated by $N_{1,est}$ and $N_{2,est}$, respectively, while the corresponding fragmented masses is indicated by $m_{1,est}$ and $m_{2,est}$, respectively.

The orbital parameters of the simulated parents are listed in Table 2.2,

Table 2.2 - Orbital parameters of the parents for the first test case.

Target					
a = 7359 km	e = 0.00348	i = 83°	Ω = 8.63°	ω = 237°	ν = 100°
Projectile					
a = 7461 km	e = 0.01459	i = 100°	Ω = 199.8°	ω = 240°	ν = 25°

with a, e, i, Ω, ω being respectively the semi-major axis, eccentricity, inclination, right ascension of the ascending node and argument of perigee.

The parent masses are M_1 = 1000 kg, M_2 = 800 kg and the impact velocity is equal to v_{imp} = 14 km/s. It is assumed that only the following masses are involved in the fragment generation: m_1 = 850 kg, m_2 = 490 kg. With these values of masses, the SBM generates a set of N_1 = 693 fragments from the target and N_2 = 453 from the projectile.

The input data for the fragmented mass are set equal to the actual parent masses, i.e., M_1 = 1000 kg, M_2 = 800 kg, while v_{imp} is set to 14 km/s. The application of the

standard SBM would produce a number of fragments, $N_{SBM,}$ about 14% and 58% more than N_1 and N_2, respectively. However, after 3 iterations, the proposed approach reduces this overestimate to 0.9% and 2.2%, respectively, while allowing the estimation of the fragmented masses with an error with respect to the assumed values of about 3% and 2%, respectively. Simulation results are summarized in Table 2.3.

Table 2.3 - Comparison between standard-SBM and iterative-SBM for the first test case.

	Parents Mass	Assumed fragmented mass	N_{cat}	m_{est}	N_{est}	N_{SBM}
Target	1000 kg	850 kg	693	875 kg	699	793
Projectile	800 kg	490 kg	453	500 kg	463	716

Figure 2.7 shows the distribution of the reference fragments in the $isin(\Omega) - icos(\Omega)$ plane. In this plane, the distance of any point from the origin is equal to the inclination of the orbit plane of the fragment, while the ratio between the x-coordinate and the y-coordinate is equal to the tangent of the RAAN. It demonstrates that, the set of fragments derived from target and projectile gather in two regions having the same inclination of the parent objects, with a certain dispersion due to the collision. This is further supported by Figure 2.8, representing the histograms of the reference and estimated fragments in the range [75, 105] degrees of inclination. Figure 2.9 represents the histograms of the reference and estimated fragments in the range [6900, 12000] km of semi-major axis. This is representative of the ΔV distribution and show how they are qualitatively similar in the two cases. This is further supported by the Gabbard diagrams in Figure 2.10 and Figure 2.11.

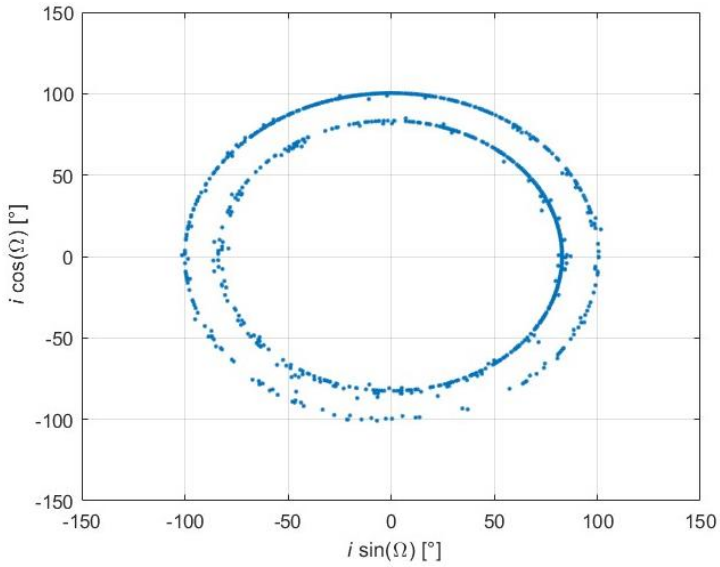
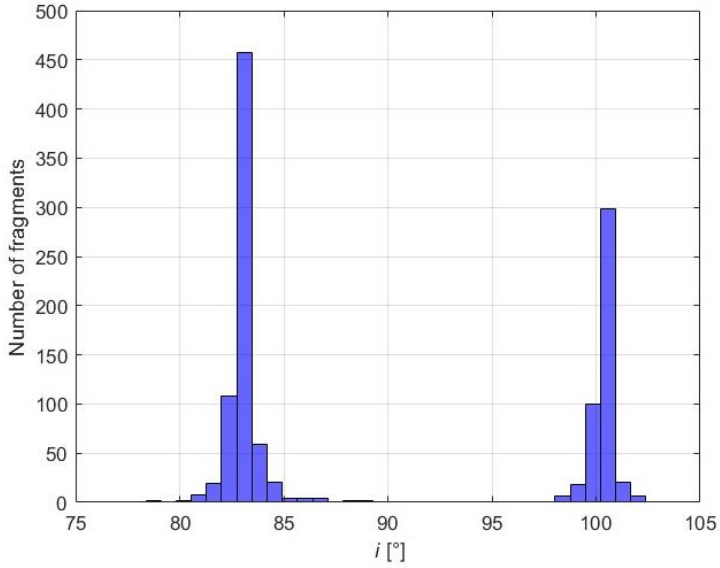
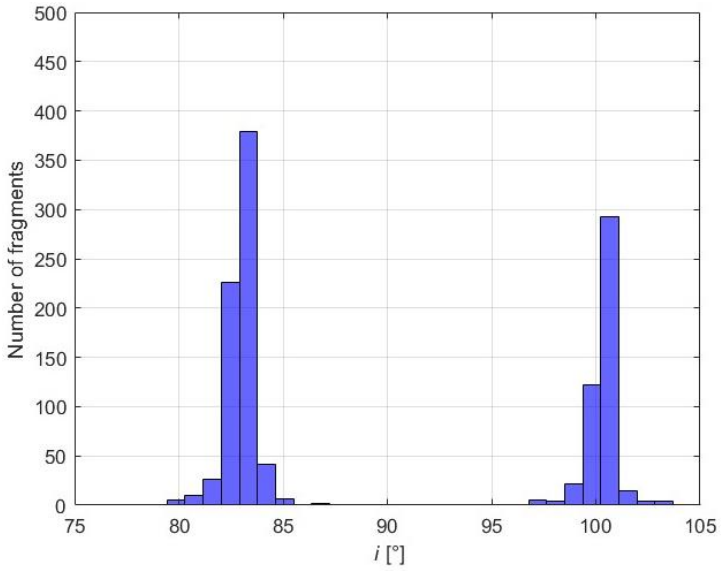


Figure 2.7 - Distribution of the reference fragments in the plane $isin(\Omega) - icos(\Omega)$ for the first test case.

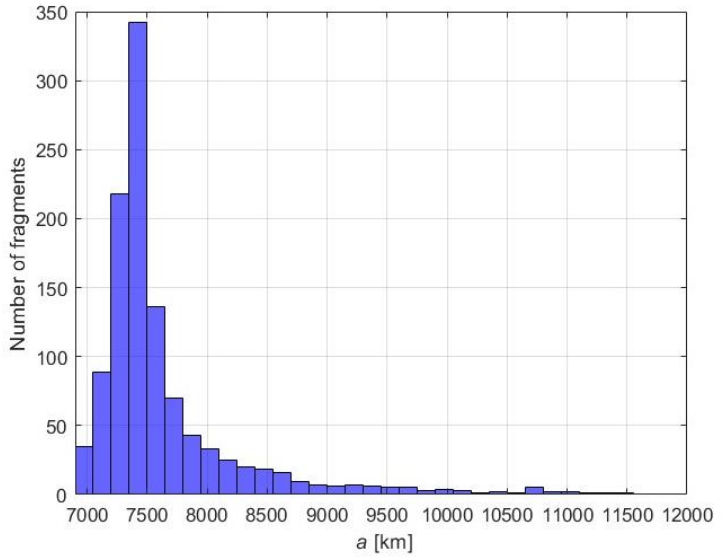


(a)

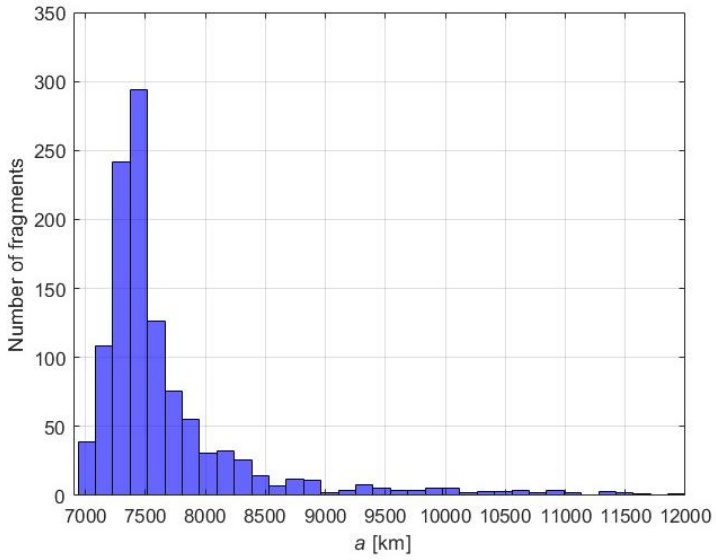


(b)

Figure 2.8 - Histograms of the reference (a) and estimated (b) fragments in the range [75°-105°] of inclination for the first test case.

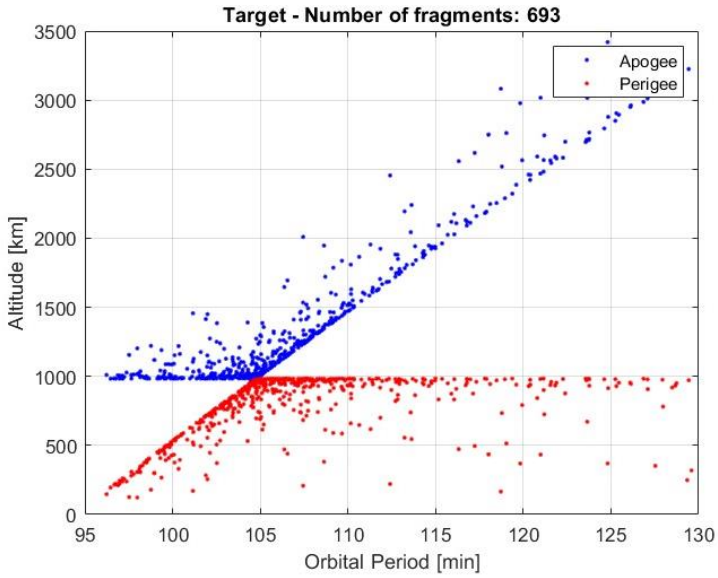


(a)

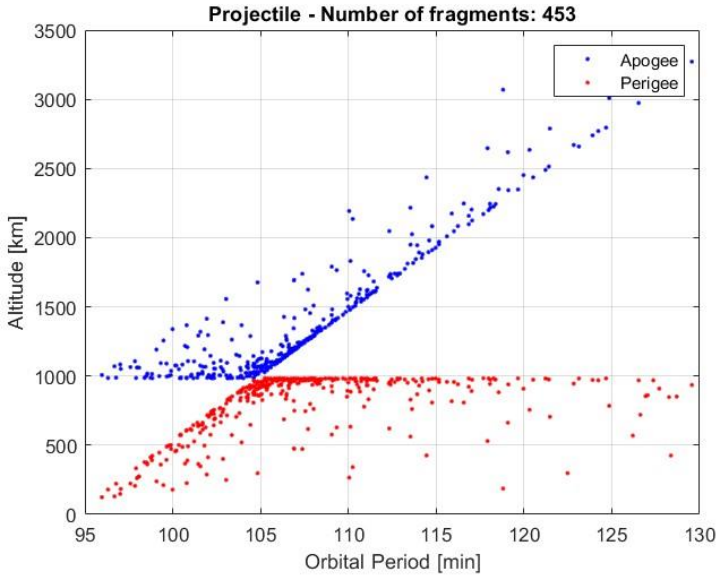


(b)

Figure 2.9 - Histograms of the reference (a) and estimated (b) fragments in the range [6900-12000] km of semi-major axis for the first test case.

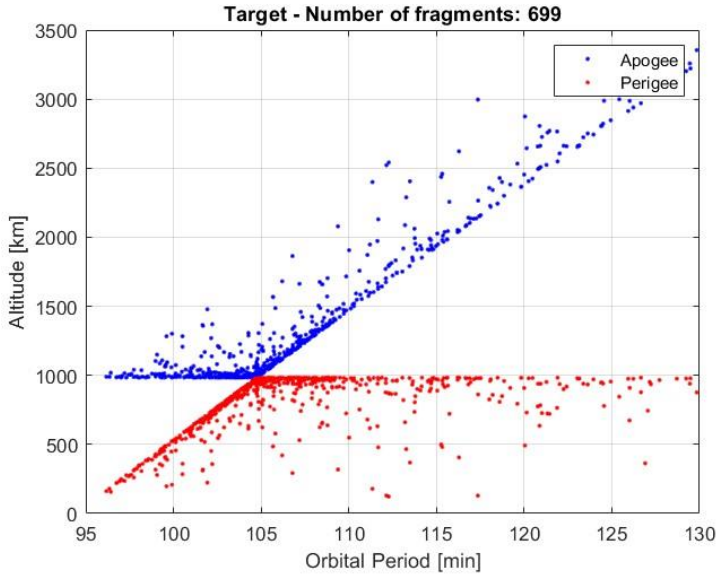


(a)

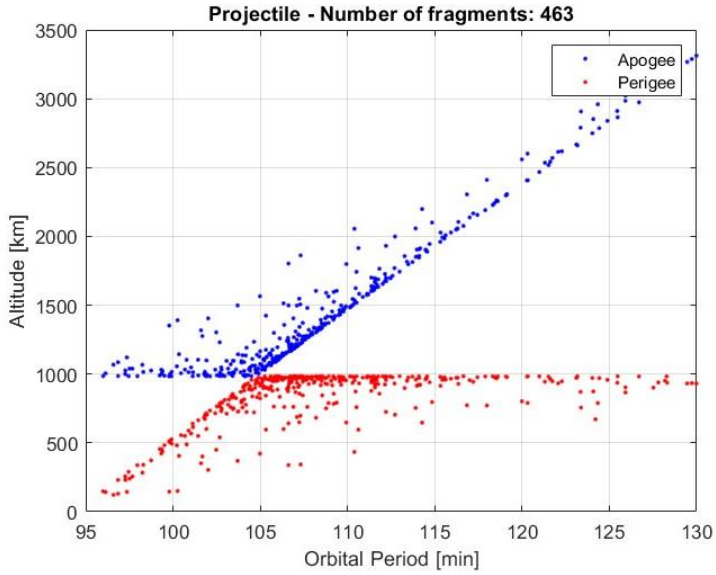


(b)

Figure 2.10 - Gabbard diagrams obtained with input masses equal to m_1 = 850 kg and m_2 = 490 kg.



(a)



(b)

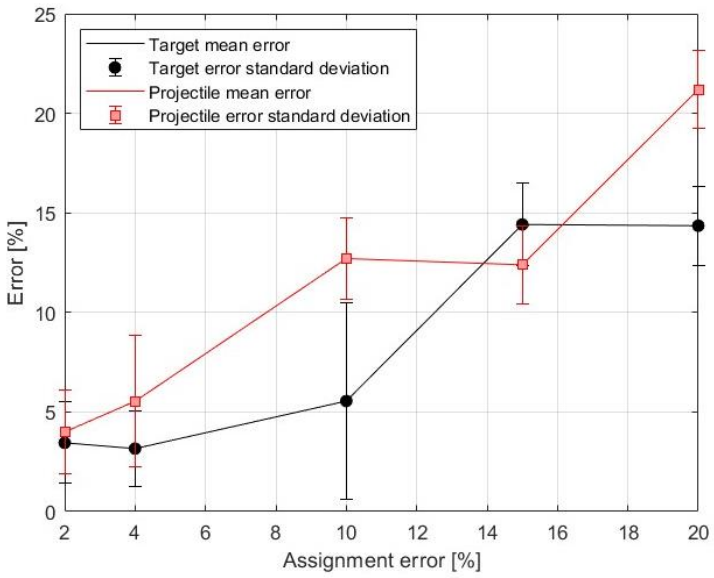
Figure 2.11 - Gabbard diagrams obtained with the fragmented masses estimated by the iterative approach for the first test case.

As stated in Section 2.1, a correct parent identification is not always guaranteed and identification issues could lead to erroneously assigned objects. To take into account this issue, an analysis of the method performance under this circumstance is carried out. It is assumed that a certain percentage of the projectile fragments has been erroneously assigned to the target. In particular, the following percentages have been considered: 2%, 4%, 10%, 15%, 20%. Table 2.4 reports the number of fragments assigned to each parent taking into account the previous percentages. The obtained number of fragments is rounded up.

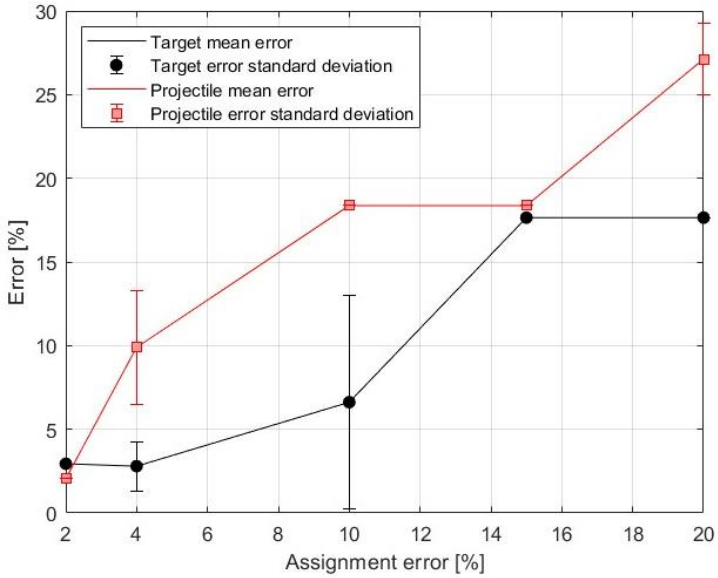
Table 2.4 - Values of N_{cat} for different percentages of assignment error.

Percentage Assignment Error	0%	2%	4%	10%	15%	20%
Target	693	703	712	739	761	784
Projectile	453	443	434	407	385	362

For each percentage assignment error, 100 runs have been executed. For each run the mean μ_{nf} and the standard deviation σ_{nf} of the number of fragments greater than 10 cm, as well as the mean μ_{mf} and the standard deviation σ_{mf} of the estimated fragmented mass, have been computed. Figure 2.12(a) shows the percentage error associated to μ_{nf} (solid line), $\mu_{nf} - \sigma_{nf}$ and $\mu_{nf} + \sigma_{nf}$ (error bars) for the target (in black) and the projectile (in red). Similarly, Figure 2.12(b) shows the percentage error associated to μ_{mf} (solid line), $\mu_{mf} - \sigma_{mf}$ and $\mu_{mf} + \sigma_{mf}$ (error bars) for the target (in black) and the projectile (in red). It must be pointed out that, for assignment errors lower than 10%, the output error is not necessarily due to the erroneous number of catalogued fragments, but also depends on the statistical variability of EVOLVE.



(a)



(b)

Figure 2.12 - (a) Mean and standard deviation errors for the number of fragments greater than 10 cm, both for target and projectile. (b) Mean and standard deviation errors for the estimated fragmented mass, both for target and projectile.

The second test case regards a non-catastrophic collision. The orbital parameters of the simulated parents are listed in Table 2.5.

Table 2.5 - Orbit parameters of the parents for the second test case.

Target					
a = 7361 km	e = 8.9 10^{-5}	i = 90°	Ω = 45°	ω = 90°	ν = 0°
Projectile					
a = 7361 km	e = 8.9 10^{-5}	i = 90°	Ω = 53°	ω = 90°	ν = 0°

The parent masses are M_1 = 1000 kg, M_2 = 50 kg and the impact velocity is equal to v_{imp} =1 km/s. It is assumed that only the following masses are involved in the collision: m_1 = 30 kg, m_2 = 30 kg. With these values of masses, the SBM generates a set of N_1 = 37 fragments from the target and N_2 = 43 from the projectile (Table 2.6), considered as reference values. The input data are then set equal to the actual parent masses, i.e., M_1 = 1000 kg, M_2 = 50 kg while v_{imp} is set to 1 km/s. Without the iterative approach, the SBM estimates fragmented masses, $M_{SBM,}$ of 50 kg for both the target

and the projectile (about 67% more than the assumed involved mass), while the number of generated fragments is equal to 68 and 74. Instead, the iterative approach allows reducing the fragmented mass estimation error down to about 17% of the assumed values. With regards to the number of fragments the SBM commits an error of 84% for the target and 94% for the projectile, while these values are reduced to about 8% and 13% by applying the iterative approach. These results are in accordance with Pardini and Anselmo [63] that highlight how the outputs of the non-catastrophic collisions are worse than catastrophic collisions in the estimation of the number of fragments. The output of the second test case is summarized in Table 2.6.

Table 2.6 - Comparison between standard-SBM and iterative-SBM for the second test case

	Parents Mass	Assumed fragmented mass	N_{cat}	M_{SBM}	m_{est}	N_{SBM}	N_{est}
Target	1000 kg	30 kg	37	50 kg	25 kg	68	34
Projectile	50 kg	30 kg	38	50 kg	25 kg	74	33

Figure 2.13 shows the distribution of the reference fragments in the $isin(\Omega) - icos(\Omega)$ plane. Differently from the first test case, the set of fragments derived from target and projectile lie on the same circle corresponding to an inclination of 90 degrees. This is further supported by Figure 2.14, representing the histograms of the reference and estimated fragments in the range [80, 100] degrees of inclination. Figure 2.15 represents the histograms of the reference and estimated fragments in the range [6900, 12000] km of semi-major axis. This is representative of the ΔV distribution and show how they are qualitatively similar in the two cases. The Gabbard diagrams in Figure 2.16 and Figure 2.17 further support this statement.

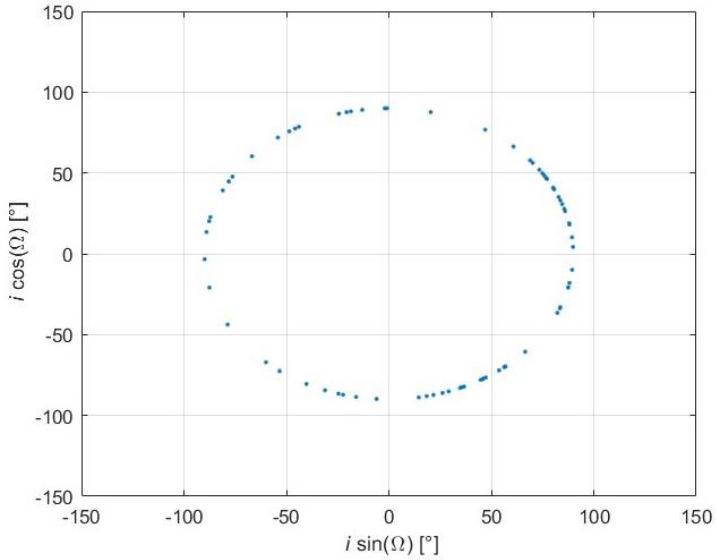
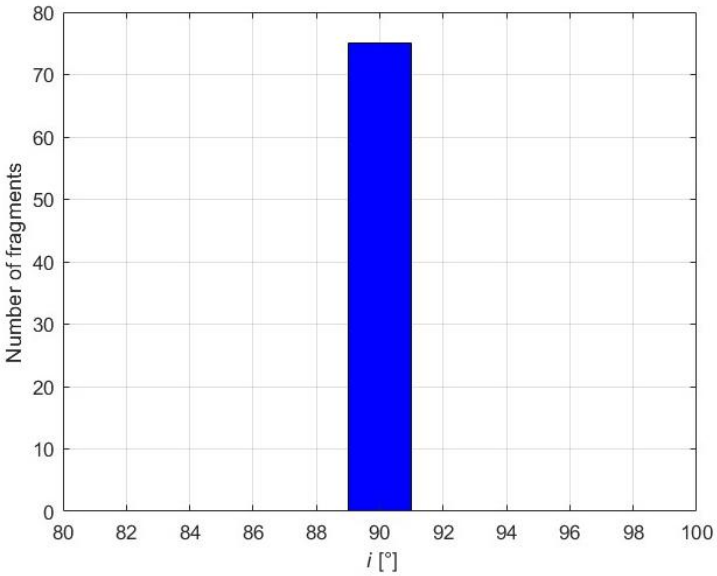
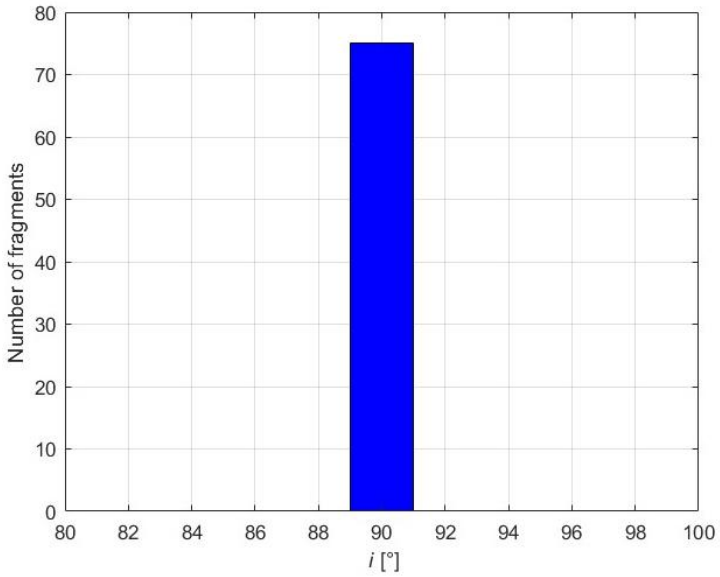


Figure 2.13 - Distribution of the reference fragments in the plane $isin(\Omega) - icos(\Omega)$ for the second test case.

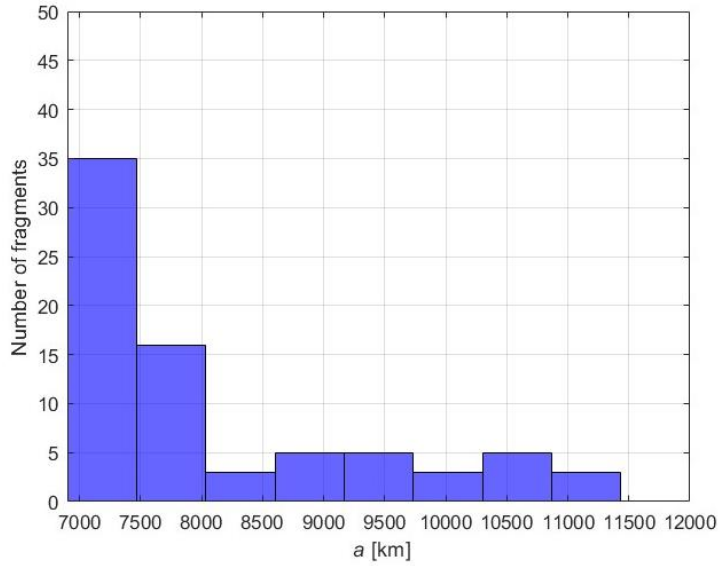


(a)

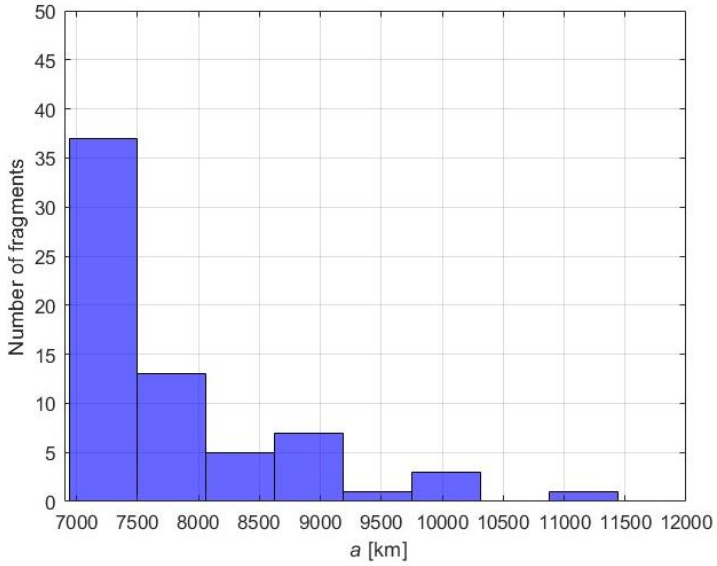


(b)

Figure 2.14 - Histograms of the reference (a) and estimated (b) fragments in the range [80°-100°] of inclination for the second test case.

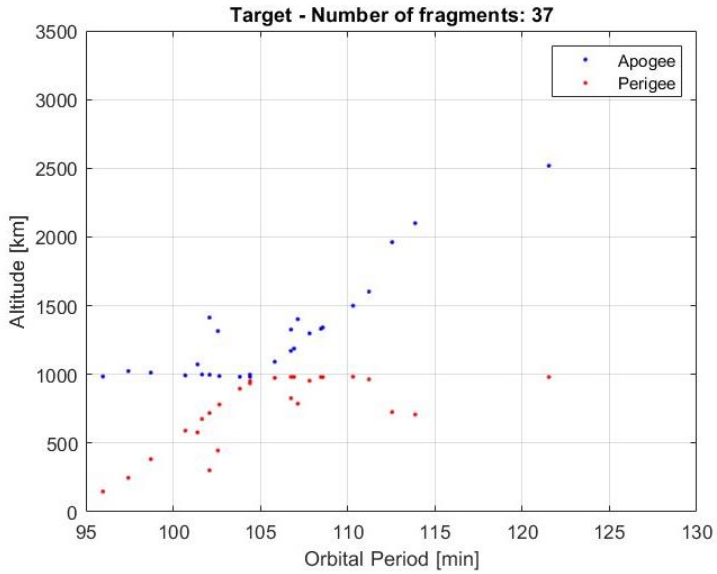


(a)

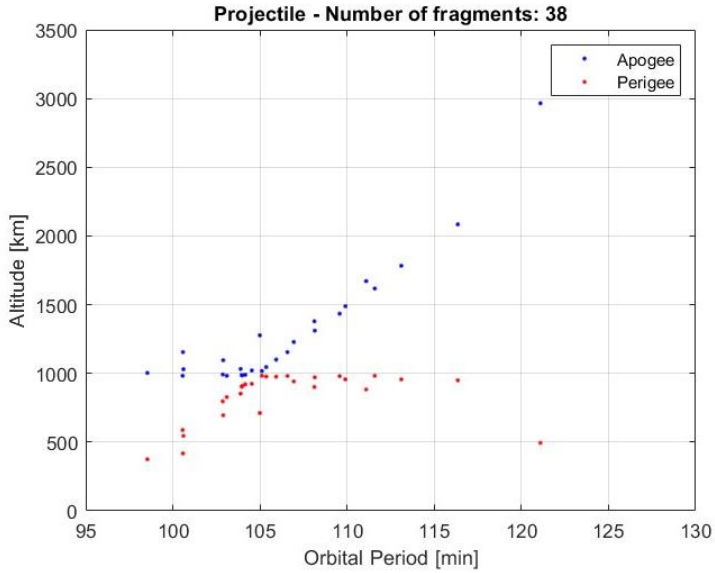


(b)

Figure 2.15 - Histograms of the reference (a) and estimated (b) fragments in the range [6900-12000] km of semi-major axis for the second test case.

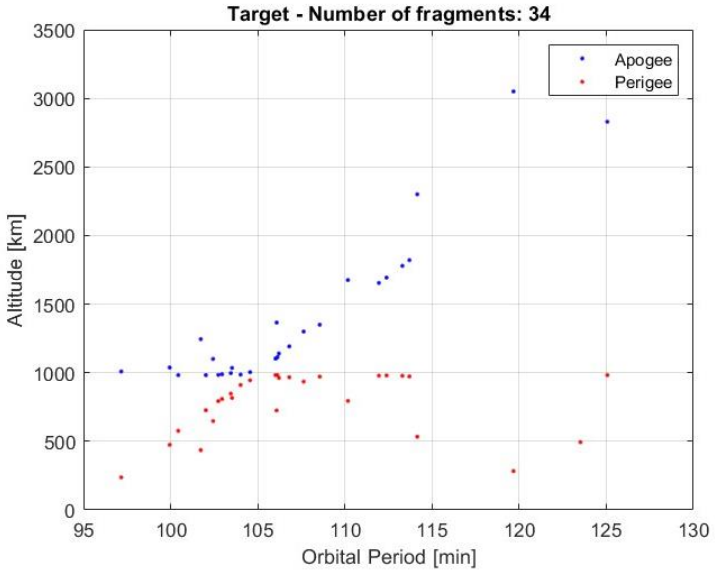


(a)

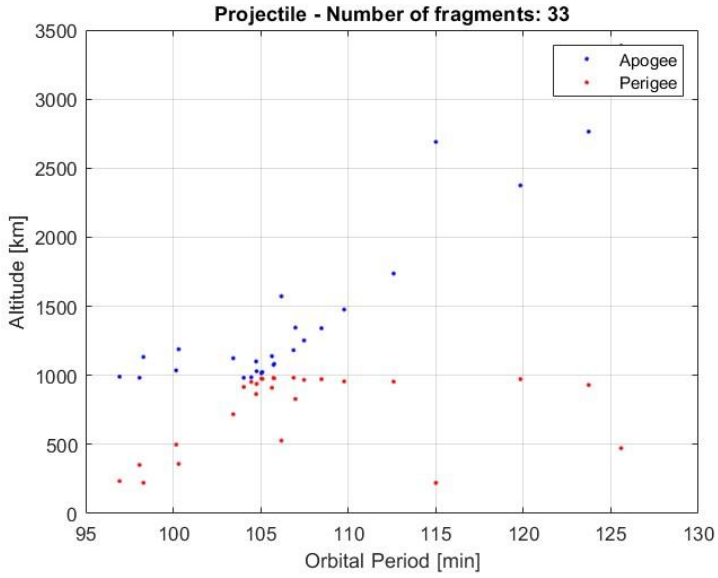


(b)

Figure 2.16 - Gabbard diagrams obtained with input masses equal to m_1 = 30 kg and m_2 = 30 kg.



(a)



(b)

Figure 2.17 - Gabbard diagrams obtained with the fragmented masses estimated by the iterative approach for the second test case.

The third test case is a real fragmentation event, namely the on-orbit collision between DMSP 5B F5 Thor Burner 2A (i.e., target) and the Catalogued debris 26207 (i.e., projectile) happened on 17 January 2005 [63]. Thor Burner 2A had a mass of 50 kg while the debris had a mass of 2.1 kg. The two objects collided at an altitude of 885 km with an impact velocity of 5.7 km/s. According to the SBM, since the energy-to-mass ratio is 62 J/g, the event is catastrophic and the number of fragments over 10 cm would be equal to 64 for the target and 0 for the projectile. However, only 6 fragments have been catalogued [63]. Hence, the SBM overestimates the number of target fragments of 967%. Assuming that all the catalogued fragments are associated to the target, the iterative approach, instead, estimates a fragmented mass of 18.75 kg for the target and 2.1 kg (i.e., the total mass) for the projectile. The number of fragments generated are $N_{1,frag}$ = 6 and $N_{2,frag}$ = 0 in accordance with the catalogue. Table 2.7 summarizes all the data.

Table 2.7 - Comparison between standard-SBM and iterative-SBM for the real test case.

	Parents mass	M_{SBM}	m_{est}	N_{SBM}	N_{est}
Target	50 kg	50 kg	18.75 kg	64	6
Projectile	2.1 kg	2.1 kg	2.1 kg	0	0

In all the previous test cases, the iterative approach has provided reasonable tuning parameters to have a number of estimated fragments within a certain pre-defined interval. It is important to highlight that, even if theoretically possible, it is not useful to further reduce the percentage error. Indeed, due to the statistical nature of the SBM and the random extraction of the fragments' characteristics, the output could have a variance greater than the pre-defined percentage error.

Some further considerations can be made with respect to the number of fragments. When the number of catalogued fragments is relatively large (e.g., more than 50 fragments), after breakup model tuning it is possible to extract from the simulated distributions sets of fragments which are consistent with the observed ones. When the number of catalogued fragments is very small, two major concerns arise. On the one hand, the type of fragmentation event (i.e., collision or explosion) may be itself uncertain. On the other hand, while it is possible to verify the consistency of observed data with theoretical distributions, the phenomenon cannot be replicated in simulations by random extractions.

Computational cost analysis

In general, the computational cost characterizing the proposed method is mostly related to the run of the SBM and so to the number of iterations required to reach convergence. For the sake of analysing the effect of the tolerance on the number of iterations, an additional simulated test case is considered. Two parents with mass equal to 800 kg collide with an impact velocity of 10 km/s. The collision is theoretically catastrophic. The algorithm is executed considering three different subcases: N_{cat} greater than 100 (i.e., 500), between 50 and 100 (i.e., 60), and below 50 (i.e., 20) for both the parents. Figure 2.18 shows the number of iterations required, for the target (solid line) and the projectile (dashed line), to reach the convergence versus a tolerance

value on N_{cat} equal to 5%, 10%, 20% and 30%. In all cases, a relatively small number of iterations is required to reach convergence.

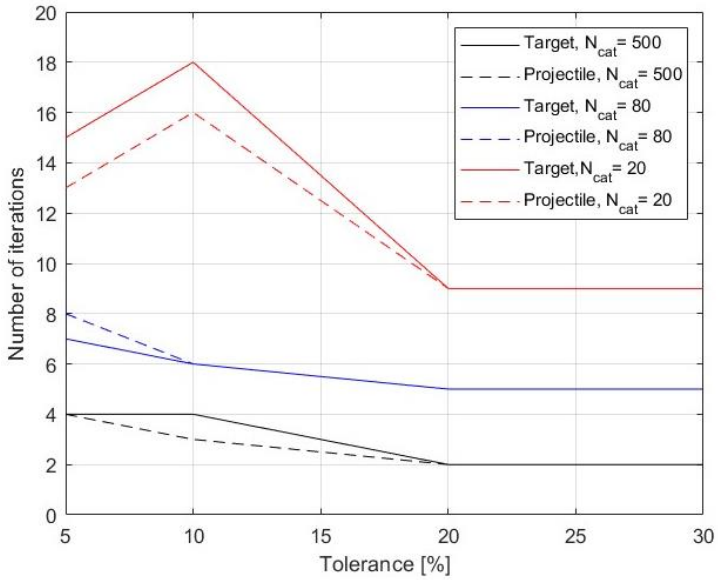


Figure 2.18 - Number of iterations against the tolerance on the number of catalogued fragments for different values of N_{cat}, both for the target (solid line) and the projectile (dashed line).

3. Resident Space Objects Characterization

This chapter tackles the problem of Resident Space Objects characterization. It provides an overview of current literature approaches to estimate several physical parameters, both physics-based and data-driven. Then, it describes the proposed Machine Learning-based methodologies, exploring different Neural Networks architectures. It presents the sensitivity analyses carried out, and the applicability of the proposed methods to real test cases. Moreover, a comparison between the ML approach and a semi-analytical methos is presented.

3.1 Literature review

The capability to characterize Resident Space Objects (RSO) is one of the main challenges for Space Situational Awareness (SSA), to support functions such as accurate orbit propagation, collision avoidance and anomaly detection. As the number of RSO is continuously growing [1], it is crucial for the safe operation of satellites to estimate and predict their trajectories. While conservative forces, including the Earth gravity force and third body perturbations, can be modelled with good accuracy, non-conservative perturbations (i.e., atmospheric drag and solar radiation pressure) may be characterized by higher uncertainty. In fact, apart from the complexity and stochastic nature of atmospheric density and solar activity modelling, these forces strongly depend on the object's characteristics which are usually either unknown or known with high uncertainty for space debris.

The features that mostly influence the trajectory of these objects are different depending on the orbital regime (i.e., LEO, MEO, GEO). Since atmospheric drag is the main non-conservative perturbation in LEO, the cross-section area, as well as the mass and the drag coefficient are the most relevant objects' characteristics on these orbits. On the other hand, due to the predominance of the solar radiation pressure (SRP) in GEO, other characteristics such as the object area exposed to the Sun, the

coefficient of reflectivity (hence, the surface material) and the mass of the object must be estimated. For this purpose, different types of sensors are used to track space objects and different types of information can be retrieved. In particular, astrometric data obtained from radars and telescopes observations provide information on the state (i.e., position and velocity) of the object, while photometric data (i.e., light curves) can be processed to provide information about the time evolution of the object's attitude and its surface material [70], [71].

Current literature proposes a wide range of approaches for RSOs' characterization depending on the type of input data and orbital regime. Badhwar et al. [24] determine the area-to-mass ratio (*AMR*) of debris in LEO using the orbital elements estimated at discrete times on the basis of radar measurements, and exploiting the inversion of the Mueller's model, minimizing a merit function by varying the *AMR*. Linares et al. [23] apply two multiple model approaches for the estimation of the *AMR* of high-AMR (HAMR) objects on geosynchronous orbits: the standard multiple-model adaptive estimation (based on angle data), and a technique called adaptive likelihood mixtures. In reference [72] an unscented Kalman filter estimation scheme using light curve and angle data is used to estimate mass, albedo-area and area of an RSO along with its associated rotational and translational states. Reference [73] presents a multiple-model adaptive estimation approach to determine the most probable shape of a resident space object in GEO among a number of candidate shape models. Linares et al. [22] employ an Adaptive Hamiltonian Markov Chain Monte Carlo estimation approach, which uses light curve data to infer the space object's orientation, shape, and surface parameters. Brack et al. [74] develop an in-orbit onboard algorithm for tracking space objects in GEO with high area-to-mass ratio. It uses a relative dynamics model and stereo-camera measurements to estimate the tracked object's position and velocity, and hence derive its solar radiation pressure coefficient (i.e., the product between the reflectivity coefficient and the *AMR*). Lacruz et al. [25] use a semi-analytical method to determine a reliability range for the *AMR* of objects in the GEO region. A reduced analytical model is validated against the General Mission Analysis Tool (GMAT) using synthetic space objects. Then, the eccentricity evolution of real RSOs is taken from CelesTrak [75] data and it is fit with a suitable evolution of the eccentricity by trying different *AMRs*. Dolado Perez et al. [76] and Águeda et al. [77] estimate the *AMR* using the

contribution of conservative and non-conservative forces to the time derivatives of the orbital parameters. Such a method is further recalled in Section 3.2.4. Finally, fitted ballistic coefficients are estimated by Pilinski M. et al. [78]. The computation involves comparing satellite drag measurements to an atmospheric model and minimizing the differences between observed and modelled drag in a special-perturbation orbit estimation scheme. The free parameter adjusted in this approach is the satellite ballistic coefficient.

In most scenarios, the estimation of the *AMR* (or, similarly, the *BC*) through these classical methods leads to better orbit prediction accuracy when compared to TLEs, or to a more realistic agreement with the actual *AMR* when this is known. However, empirical methods suffer from sensitivity to noise and outliers in TLE series, which can significantly degrade estimates of the *AMR*. Batch least-squares and sequential filters [79], [80], [81], [82], [83], [84] tend to be more robust; however, they usually place restrictive assumptions on the dynamics and measurements (such as additive white process noise), and they require a fair amount of tuning when noise characteristics are unknown. In addition, the full state of the RSO usually needs to be estimated concurrently to the *AMR*, and decoupling the *AMR* from uncertainty in the atmospheric density needs to be performed by implementing high-fidelity atmospheric models [85]. These factors make a robust estimation of the ballistic coefficient computationally intensive, which complicates their scalability to large sets of objects.

Machine learning methods based on neural networks (NNs) can ameliorate some of these drawbacks, and have recently been applied to the prediction of re-entry trajectories [86] and to the simultaneous orbit propagation and estimation of the drag coefficient [87]. Specifically, NNs do not impose any qualitative assumption on the noise affecting observations, which can be decoupled and correlated. In principle, decoupling of atmospheric uncertainties from changes in *AMR* can be directly achieved by including the spacecraft state, epoch, and space weather drivers as features, thus avoiding the need for complex atmospheric models during estimation. Finally, once a NN has been trained on an appropriate dataset, its evaluation is computationally efficient. This is particularly significant as typical NNs for estimation problems in astrodynamics have a limited number of parameters.

Several works have applied NNs and, in general, ML techniques to space objects characterization. Howard et al. [88] propose a machine learning technique for feature recognition (e.g., shape and orientation) of RSO in the geosynchronous regime. A set of simulated light curves is generated varying geometry, orientation, material and attitude dynamics conditions (e.g., three-axis/spin stabilised, tumbling). Based on a sequence of one or more measurements, a classification algorithm is able to recover these characteristics by analysing photometric signatures. Similarly, Linares et al. [28] use a deep Convolutional Neural Network (CNN) to extract features from brightness measurements, considering physics-based models that account for rotational dynamic and light reflection properties of RSOs. Furfaro et al. [89] investigate the problem of training deep neural networks architectures to discriminate between different RSO using light curves measurements and explore the meta-learning approach as a possible training strategy. Their work demonstrates that CNNs can be trained both on simulated and real light curves observations to classify space objects and that meta-learning may provide a framework for learning using a very limited amount of data. Kerr et al. [90], [91] develop a classification algorithm that retrieves from light curves information regarding shape and size of GEO objects using several Artificial Intelligence (AI) techniques (i.e., autoencoders, Convolutional Neural Networks and Recurrent Neural Networks). Vasile et al. [92] propose a classification system for unidentified satellites by processing hyperspectral sensor data. They analyse the time-varying spectrum of a space object as received from a ground-based hyperspectral sensor (which is function of the surface material of the space object), and extract information regarding its materials, the components that may be on board (e.g., solar panels and antennas), and finally classify the object. Moreover, in [93] they demonstrate the use of ML to estimate the attitude motion by associating time series of spectral responses to the corresponding quaternion time history. Badura et al. [94] develop an autoencoder-based anomaly detection algorithm that exploits light curves to provide spin-stability assessment. The algorithm includes unevenly time-sampled data in the training phase, as well as geometry and measurement uncertainty information in the ML model. Qashoa and Lee [95] present a novel classification approach for LEO objects that estimates the type of object (i.e., stable satellite, tumbling satellite, and rocket body) and spin rate from light curves using a wavelet scattering transformation, with both a

conventional ML approach (i.e., Support Vector Machine), and a deep learning technique (i.e., Long Short-Term Memory). Enriquez and Amato [96] use Long-Short Term Memory recurrent neural networks to retrieve the attitude motion from light curves of RSOs in LEO, assuming that the geometry, construction materials and orbit of the target object are known. Peng and Bai [97] adopt the random forest method to recover the area-to-mass ratio, using the consistency error as defined in [98], [99]. Their work uses a simulation-based space catalogue environment, focusing on Sun-synchronous orbits in LEO.

3.2 Feedforward Neural Networks

In the framework described in the previous section, this PhD thesis further investigates the possibility to use ML-based approaches to recover the ballistic coefficient (*BC*), that is the product between the drag coefficient and the *AMR*, from astrometric data. Unlike existing literature, a wide range of orbital parameters in LEO are considered. Several ML techniques are compared and a sensitivity analysis based on the number of objects used for the training set and on the number of trainings is carried out. This analysis has thus the aim to assess dataset requirements and tuning options to effectively support ML-based RSO characterization, which represents another important innovation point. A simulation environment for the training, validation and testing phase is used. Once the best performing ML-technique is identified based on *ad-hoc* defined performance metrics, results are shown which come from the application of such technique to real-world test cases, going beyond simulation-based analyses.

It is important to make the following consideration. Several recent works have developed accurate models and algorithms to estimate the different parameters composing the atmospheric drag: the atmospheric density [83], [100], the drag coefficient [101], [102], [103], [104], [105], [106], and the area-to-mass ratio as mentioned above. However, each of these parameters can be estimated only with a certain accuracy due to the stochastic nature of the space weather and the intrinsic

measurements errors in both astrometric and photometric data. In general, errors coming from *BC* mismodeling and atmospheric density misprediction are not easy to be separated since they have the same effect (i.e., a linear dependence) on the drag equation. This means that the global error can be simulated by introducing uncertainties in only one of the inputs. In this perspective, since the proposed techniques are based on a given atmospheric model (i.e., Jacchia-Bowman) and on a constant value of the drag coefficient, the uncertainty in the atmospheric density and drag coefficient modelling enter the estimation process and will impact the *BC* prediction. Hence, the proposed method estimates a "fitted" *BC*. Nevertheless, a method to improve atmospheric modelling by including space weather indices in the training process is also discussed.

3.2.1 Methodology

Data mining and machine learning techniques refer to the methods and algorithms that can discover patterns and generate new information by analysing existing data [97]. In a typical ML technique, an outcome measurement must be predicted based on a set of features. A training dataset is available in which it is possible to observe the outcome and feature measurements for a set of objects. Using these data, a prediction model, or *learner*, is built which will allow the prediction of the outcome for new unseen objects. A subset of the training data is used to improve the learner in the validation phase. Then, a testing dataset is used to evaluate the learner performance receiving in input new unseen information [107].

In this work, the training set consists of orbital data from a number of synthetic space objects with known *BC*. To build such training set, the following steps have been followed. First, discrete values of the *AMR* in the range [0.01, 0.6] m^2/kg have been defined as reported in Table 3.1. Such interval has been chosen for two reasons. First, we have started from the range considered in current literature based on ML-methods [97], equal to [0.02, 0.3] m^2/kg, and have extended it. Second, considering the usage of the approach within fragmentation analyses, most of the fragments generated by

past breakup events (i.e., collisions and explosions) have a value of *AMR* greater than 0.01 m^2/kg [108] and experiments of hypervelocity impacts of new smaller satellites confirm this range [58], [109]. Hence, this range seems a good starting point to test the observability of the *BC* on the orbital parameters, and hence, the performance of the ML algorithm. Specifically, the step between two consecutive *AMR* values is finer for lower values of *AMR* to better fit the effect of small variations of the *BC* on the propagation. Similarly, discrete values of altitude (and hence of the semi-major axis) in the range of [400, 600] km have been defined (see Table 3.1). For the other orbital parameters, *ad-hoc* intervals have been considered. As regards the inclination, an interval of [30, 110] °, where most of catalogued LEO objects lay, has been defined. Eccentricity values in the interval [10^{-4}, 10^{-2}] and values of right ascension of the ascending node, argument of perigee and true anomaly in the interval of [0, 360] ° have been considered. For each possible combination of *AMR* and semimajor axis, a set of 10 objects is generated by randomly extracting their orbital parameters from uniform distributions inside the aforementioned intervals. Each of the $N_{obj,tr}$ objects thus generated has its own set of orbital parameters and its own *AMR*.

Table 3.1 - Interval definition for training set data.

Parameter	Value
AMR	[0.01, 0.02, 0.03, 0.05, 0.07, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.60] m^2/kg
Altitude, h	[400, 420, 440, 460, 480, 500, 520, 540, 560, 580, 600] km
Eccentricity, e	[10^{-4}, 10^{-2}]
Inclination, i	[30, 110] °
Right Ascension Ascending Node, Ω	[0, 360] °
Argument of Perigee, ω	[0, 360] °
True Anomaly, ν	[0, 360] °

As regards the value of the drag coefficient (C_d), it is important to highlight that the C_d of a resident space object depends on several factors including its geometric shape, and it can be highly variable due to attitude shifts and/or variations in the atmospheric conditions (such as molecular composition and ambient temperature) which can respectively produce changes in the angle of the flow with respect to the surface and in the object wall temperature. In particular, the C_d can range from 0.4 to 4.3 in the altitude range of interest in this work [110]. In this respect, the proposed approach is aimed at the estimation of the ballistic coefficient (i.e., the product between the drag coefficient and the area-to-mass ratio). However, for the sake of simplicity in the generation of training and testing datasets, a fixed value to C_d (equal to 2.2) has been set in order to vary only the area-to-mass ratio. Such strategy actually includes the uncertainties in the true value of the C_d within the variable *AMR* coefficient. Therefore, a ballistic coefficient *BC* will be assigned to each synthetic space object. Each object is propagated twice for a time interval of 30 days: the first propagation uses the actual value of the *BC* (and thus is referred to as *true* propagation) while the second propagation uses a default value of *BC* equal to 1 m^2/kg (referred to as *false* propagation). Both propagations are carried out using the same orbital propagator in terms of perturbations' modelling, thus isolating only the impact related to the *BC*. In particular, a numerical orbital propagator is adopted that integrates and solves the equation of motion in Cowell's formulation by using a Runge-Kutta scheme of 8th-9th order. The following perturbations are considered: 20 x 20 harmonics of the EGM2008 gravity model [111], the Jacchia-Bowman 2008 atmospheric model [112], Sun and Moon as third-body perturbation and a spherical model for Solar Radiation Pressure (SRP) [113]. Space weather is accounted for using data from CelesTrak [75]. In particular, the Jacchia-Bowman atmospheric model uses the planetary range index (K_p). Starting from the initial epoch, propagation data (i.e., position and velocity) are then sampled with a frequency of one measurement per day to take into account both the average number of available measurements per day for LEO objects [114] and a minimum time to provide observability for the non-conservative effects. The difference between the state vectors obtained with the *true* and *false* propagation at each considered propagation time Δt, hereinafter referred to as consistency error $\boldsymbol{\delta X_{\Delta t}}$, is computed as:

$$\boldsymbol{\delta X_{\Delta t} = X_{BC1} - X_{BC}} \tag{3.1}$$

where $\boldsymbol{X_{BC1}}$ is the state vector obtained with the *BC* equal to 1 m^2/kg, $\boldsymbol{X_{BC}}$ is the state vector obtained with the actual *BC*. Therefore, a set of 30 consistency errors can be associated to the same object corresponding to the 30 sampled measurements. From the state vector, containing position and velocity expressed in Earth Centred Inertial (ECI) reference frame, it is possible to derive the osculating orbital parameters, position and velocity in RSW Reference Frame [113], and the Brouwer mean orbital elements corresponding to that propagation time [113]. As it will be shown later in this section, these quantities can be likewise introduced as components of the state vector. For the validation phase, a certain percentage of the training set is used. According to literature, this value has been set equal to 20% [115].

The Machine Learning algorithm will learn to map the consistency error to the real value of the ballistic coefficient. The overall architecture of the training and validation phase is shown in Figure 3.1.

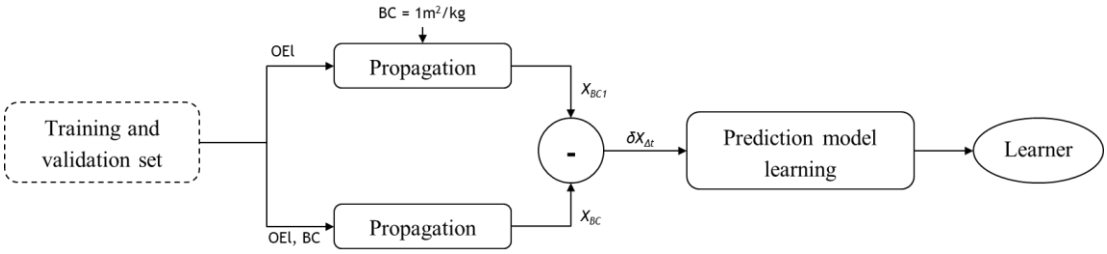


Figure 3.1 - Architecture of the training and validation phase.

It is important to observe that, in this work, the *BC* is assumed to be constant over the time of interest. For the vast majority of Earth orbiting objects, within the time scale considered for the analyses, the mass is approximately or exactly constant, and the frontal area variations are limited thus having a small impact if compared with density modelling errors [116], [117]. Moreover, as regards space debris, there are few considerations to be made based on their typical shapes. In case of spherical objects,

there is no variability of the *BC* due to the symmetry. In case of elongated or plate-shaped objects, two scenarios can be expected:

1) tumbling objects: there is certainly a variation of the *BC*; however, the method will estimate a mean *BC* also considering that the tumbling period is typically shorter than the interval considered for the analyses

2) objects that exhibit a (slow) continuous increase/decrease over the interval of interest: in this case, the proposed algorithm will estimate the mean *BC* within the considered time interval.

In addition, it is worth highlighting that a variation in the value of BC should be large enough to generate observable impact on the evolution of orbital parameters.

A testing set of data is generated to evaluate the ML-technique performance. To this aim, the same intervals for orbital parameters and *AMR* of the training set have been considered. A set of $N_{obj,te}$ objects has been generated with the following characteristics: the *AMR* and the orbital parameters have been randomly extracted from uniform distributions in the intervals reported in Table 3.1. Each object has been propagated twice (i.e., using a constant *BC* of 1 m^2/kg, and the true *BC* values, respectively), with the same propagation settings of the training set, for 30 days. Clearly, the propagations with the true *BC* values are carried out to reproduce the availability of astrometric data as provided by ground sensors. Propagation data have been sampled with a frequency of one measurement per day. For each object and for each time interval, a consistency error obtained as a difference between the *true* and *false* propagation can be computed and given as an input to the ML algorithm. The choice of propagating the same object for different time intervals is further supported by the fact that real RSOs have several Two-Line-Elements (TLE), each of them corresponding to a different epoch. In this way, the testing set will simulate the actual application of the ML algorithm to real cases as described in the following section. The overall architecture of the testing phase is shown in Figure 3.2.

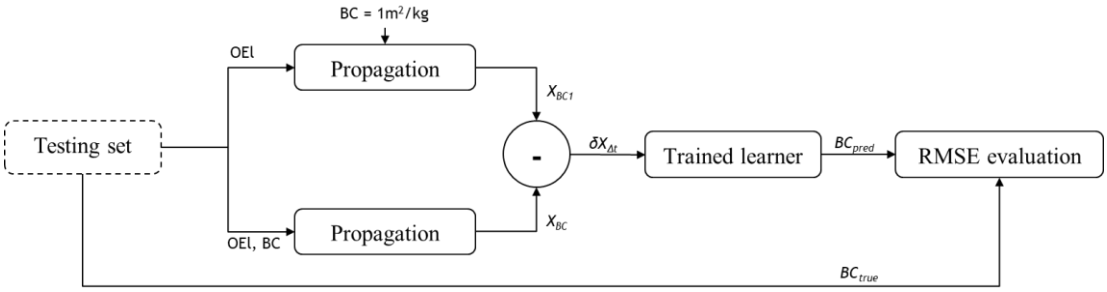


Figure 3.2 - Architecture of the testing phase.

The root mean square error (*RMSE*) between BC_{pred} and BC_{true} is used as performance metric, and is computed as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(BC_{pred,i} - BC_{true,i}\right)^2} \tag{3.2}$$

where N is the total number of predictions made (i.e., the total number of objects of the testing dataset), $BC_{pred,i}$ and $BC_{true,i}$ are the predicted and true *BC* of the i-th object of the testing dataset.

When the algorithm is applied to real-world test cases, Two Line Elements (TLEs) are considered as reference values. For each RSO to be characterized, all the available TLEs over a time interval of 30 days are taken from a space catalogue [114]. Before being used for the object characterization, the TLEs undergo a filtering process aimed to remove eventual outliers or repeated information. The process consists of two steps:

- whenever there are two or more TLEs corresponding to the same epoch, only the last one is considered

- the orbital elements are expressed in an equinoctial space (Eq. 3.3) as their temporal evolution is smoother and thus better suited for outlier detection and filtering in absence of any external information [76]:

$$\begin{pmatrix} a \\ e_x \\ e_y \\ i_x \\ i_y \\ \lambda \end{pmatrix} = \begin{pmatrix} a \\ e\cos(\omega + \Omega) \\ e\sin(\omega + \Omega) \\ \sin(i/2)\ \cos(\Omega) \\ \sin(i/2)\ \sin(\Omega) \\ \omega + \Omega + M \end{pmatrix} \tag{3.3}$$

where a is the semimajor axis, e the eccentricity, i the inclination, Ω the right ascension of the ascending node (RAAN), ω the argument of perigee, and M the mean anomaly. Moreover, in order to avoid the sinusoidal evolution, the second and the third parameters are combined, and just the eccentricity evolution is considered for the filtering of the outliers. Hence, a polynomial fitting to detect outliers is performed over the semi-major axis, the eccentricity and the inclination vector (i.e., i_x and i_y in Eq. 3.3). In particular, a 3rd order fitting polynomial is determined using all the TLEs in the considered time window. Then, setting a tolerance tol of 15%, if Eq. 3.4 is satisfied the i-th TLE is considered as outlier:

$$\left|\frac{OEl_{TLE,i} - OEl_{P,i}}{OEl_{P,i}}\right| 100 > tol \tag{3.4}$$

where $OEl_{TLE,i}$ is the orbital element (i.e., a, e, i_x and i_y) obtained from the i-th TLE, $OEl_{P,i}$ is the orbital element (i.e., a, e, i_x and i_y) corresponding to the i-th TLE evaluated with the fitting polynomial.

After the filtering process, the space object is propagated (using the numerical propagator described before) with a ballistic coefficient equal to 1 m^2/kg, having as initial orbital parameters the ones corresponding to the first TLE. The orbital elements, thus obtained, are then sampled with a frequency of one measurement per day and converted into mean orbital elements. Afterwards, using spline interpolation, the values of the orbital elements at the epochs of the TLEs are computed and Eq. 3.1 can be applied. Figure 3.3 shows the overall architecture of the algorithmic application to real RSOs.

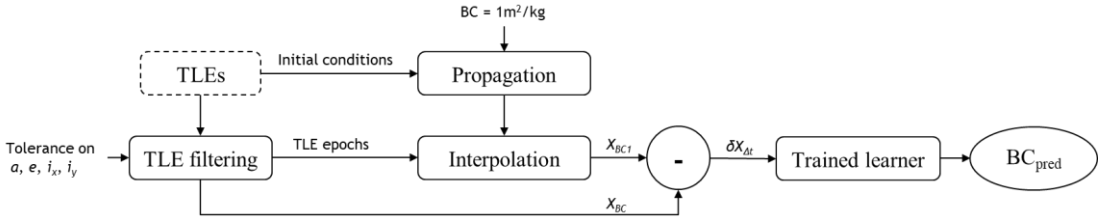


Figure 3.3 - Architecture of the algorithm application to real RSOs.

Features description and selection

Three main types of ML techniques have been explored, all of them available on the MATLAB Regression Learner App [118]: Regression Trees (i.e., Fine Tree, Medium Tree and Coarse Tree), Ensemble of Trees (i.e., Boosted Trees and Bagged Trees), and Neural Networks (NN) (i.e., Narrow, Medium, Wide, Bilayered and Trilayered). The training and testing sets used for all the techniques are the same. For each ML technique, different features can be selected to train the learner. The choice of the selected features is crucial for the learning process as they must capture the relation between the outcome and the input data. In this work, 31 features have been explored:

- initial orbital elements (OEl): a (semimajor axis), e (eccentricity), i (inclination), Ω (RAAN), ω (argument of perigee), ν (true anomaly)
- initial position and velocity in ECI, $(XYZ)_{ECI}$ and $\left(V_x V_y V_z\right)_{ECI}$ respectively
- the difference between the true and false orbital parameters (δOEl) at the end of the propagation: $\delta a, \delta e, \delta i, \delta\Omega, \delta\omega, \delta\nu$
- the difference between true and false position and velocity in ECI at the end of the propagation: $\delta(XYZ)_{ECI}$ and $\delta\left(V_x V_y V_z\right)_{ECI}$ respectively
- the difference between true and false position and velocity expressed in RSW Reference Frame [113] with the origin in the true object at the end of the propagation: $\delta r, \delta s, \delta w, \delta v_r, \delta v_s, \delta v_w$

- the time interval Δt between the initial epoch and the epoch at which the set of consistency errors refers to.

To choose the set of features that minimizes the RMSE, different combinations of such features have been examined. In particular:

- Set # 1 refers to all the 31 features listed above
- Set # 2 refers to in-plane features and propagation time: $a, \delta a, \delta r, \delta s, \delta v_r, \delta v_s, \Delta t$
- Set # 3 adds to Set # 2 the initial argument of perigee ω and true anomaly v and their variation between true and false propagations $\delta\omega$ and δv. This is to consider the actual position of the satellite on the orbit
- Set # 4 adds to Set # 3 the initial position in ECI $(XYZ)_{ECI}$and its variation between true and false propagations $\delta(XYZ)_{ECI}$
- Set # 5 includes the orbital elements OEl, the variation δOEl, the consistency errors in RSW ($\delta r, \delta s, \delta w, \delta v_r, \delta v_s, \delta v_w$) and the propagation time Δt
- Set # 6 adds to Set # 5 the initial position in Earth-Centered Inertial (ECI) reference frame, $(XYZ)_{ECI}$ and its variation between true and false propagations $\delta(XYZ)_{ECI}$
- Set # 7 considers initial orbital elements OEl, the variation δOEl and the propagation time Δt

Another important consideration regards the way to express the orbital elements and the respective variations δOEl. In fact, osculating orbital elements derived from the propagation present significant short-term oscillations that might mislead the prediction model. Therefore, both osculating and mean elements have been considered. Figure 3.4 reports the RMSE obtained for each set of features, considering all the available ML techniques and the simulated testing set, both for osculating (a) and mean (b) orbital elements. The following acronyms are used in the legend: NNN (Narrow Neural Network), MNN (Medium Neural Network), NNN (Wide Neural

Network), BNN (Bilayered Neural Network), TNN (Trilayered Neural Network), FT (Fine Tree), MT (Medium Tree), CT (Coarse Tree), BoT (Boosted Trees), BaT (Bagged Trees). Some considerations can be done: 1) for both osculating and mean orbital elements the neural networks outperform in most of the cases the Regression Trees and the Ensemble of Trees; 2) there is not a specific neural network that performs better than the others for all the feature sets and the minimum RMSEs obtained exhibit similar values; 3) in all the cases, the RMSE is minimised when mean orbital elements are considered; in particular, for a fixed set of features, there is a reduction of the RMSE ranging from 3% to 30% when considering mean orbital elements instead of the osculating ones. This is due to the fact that short term oscillations, as expected, hide the effect that the ballistic coefficient has on the propagation and prevent the algorithm to learn the prediction model.

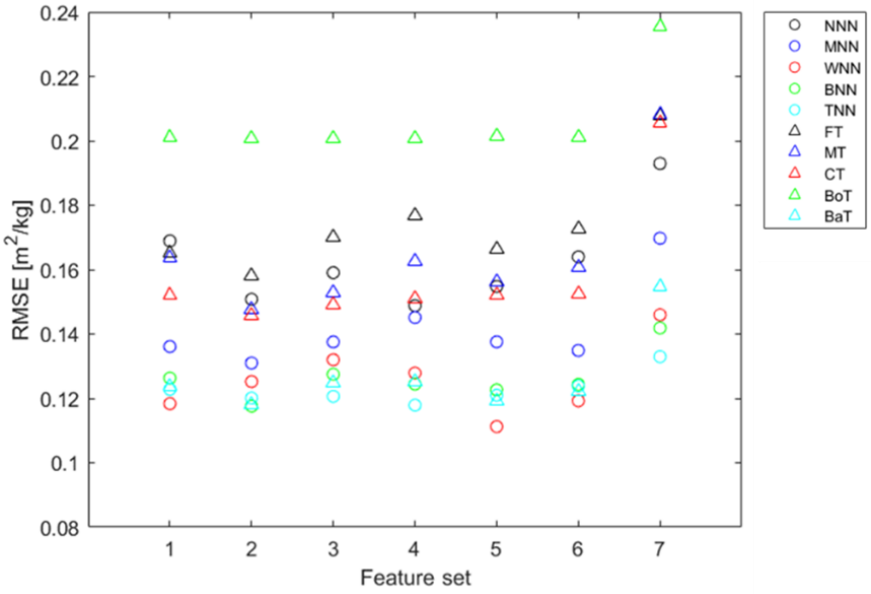
In this phase of the work, another performance metric has been adopted to support the considerations made, i.e., the RMSE of the relative errors (RMSRE), defined as follows:

$$RMSRE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{BC_{pred,i} - BC_{true,i}}{BC_{true,i}}\right)^2} \tag{3.5}$$

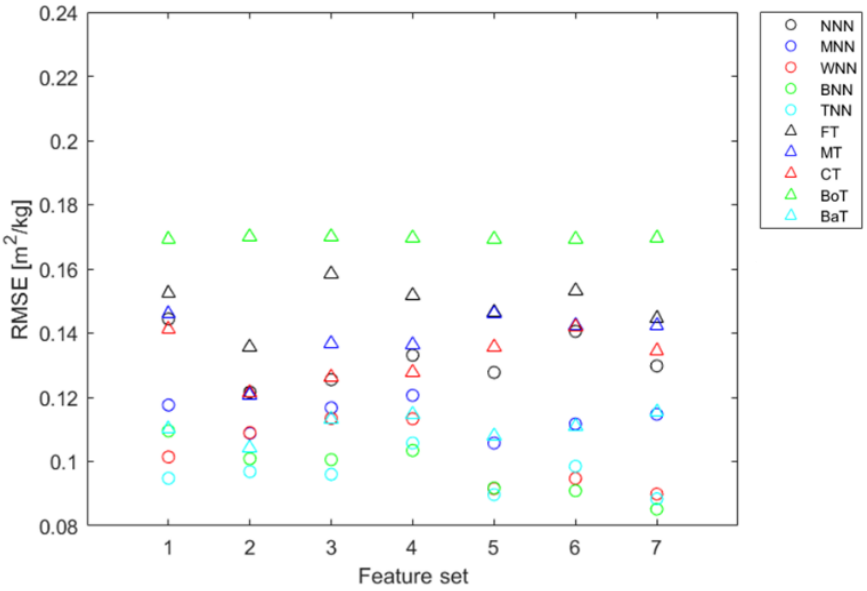
Using this metric, the same aforementioned considerations can be made. Table 3.2 reports the computational cost necessary to predict the *BC* of the whole testing data set, for each ML technique and for one set of features (# 7). Similar performance has been obtained with the other sets of features. All the simulations have been run using MATLAB R2021b on a desk workstation equipped with an Intel Core i9-11900 processor operating at 2.5 GHz and 32GB of RAM. It can be noticed that the Regression Trees and the Ensemble of Trees need a lower computational time, at the cost of a lower accuracy.

Table 3.3 reports the minimum RMSE obtained for a specific set of features, considering all the available ML techniques and the simulated testing set. The ML technique that minimises the RMSE is indicated in the third column. Also, the use of mean (M) or osculating (O) parameters is reported. For all these reasons, in the

remainder of this work, analyses will be performed considering only neural networks, mean orbital elements and feature set # 7 as it has provided the minimum RMSE among the analysed ones.



(a)



(b)

Figure 3.4 - RMSE for each set of feature and ML-technique when considering osculating (a) and mean (b) orbital elements.

Table 3.2 - Computational cost to predict the *BC* of the whole testing data set, for each ML technique and for feature set # 7.

ML tech	NNN	MNN	WNN	BNN	TNN	FT	MT	CT	BoT	BaT
Time [s]	9.9	15.7	49.5	17.9	25.1	1.0	0.4	0.1	1.3	2.8

Table 3.3 - Feature set performance – The first column reports the feature set number. The second and third columns report the minimum RMSE and the ML-technique that minimizes the RMSE respectively. The last column reports the type of orbital parameters: mean (M) or osculating (O).

Feature set	Minimum RMSE [m^2/kg]	Best ML-Technique	Mean (M) / Osculating (O)
# 1	0.095	TNN	M
# 2	0.097	TNN	M
# 3	0.097	TNN	M
# 4	0.103	BNN	M
# 5	0.090	TNN	M
# 6	0.091	BNN	M

3.2.2 Sensitivity analysis and performance in simulated test cases

In this section, a sensitivity analysis is carried out. The behaviour of the neural networks has been analysed with respect to the number of trainings, the propagation time, the number of objects of the training data set, and the frequency of the measurements. The training and testing sets consist of 1650 and 50 objects respectively, generated as described in section 2.1 and 2.2. The initial epoch for the synthetic objects' propagation is the 1st January 2021 at 00:00:00.000 UTC. This epoch

has been chosen to have access to a big amount of publicly available updated data. For the interval of interest, the K_p index is in the range of [0, 4.70].

Number of trainings

Each neural network can predict, for fixed training and testing data sets, different values of *BC* if trained N_{tr} times, due to the random initialization of weights [119]. Moreover, as mentioned in Section 3.2.1, N_{days} measurements (i.e., one measurement per day) are assigned to each of the $N_{obj,te}$ objects. Therefore, for each RSO of the testing set N_{days} x N_{tr} different values of *BC* can be predicted. For each day, the median of the N_{tr} predictions is computed and, afterwards, the median of the values relative to the N_{days} days is calculated. In this way, a *BC* is estimated for each RSO, hereinafter referred to as BC_{pred}. Median values are preferred to mean values to neglect the effect of outliers. To evaluate the effect that the number of trainings has on the performance of the neural networks, the RMSE is computed for every neural network using Eq. 3.2, varying the number of trainings from 5 to 100. Figure 3.5 shows the RMSE as a function of the number of trainings. The WNN exhibits, in all the cases, the smallest RMSE. Moreover, the MNN, the WNN and the TNN do not show a significant variation of the RMSE with the number of trainings. The NNN and the BNN instead have a larger and smaller RMSE for N_{tr} greater than 30 respectively. Considering the RMSE and the computational cost, that is linear with respect to the number of trainings, for the following analyses N_{tr} has been set equal to 10.

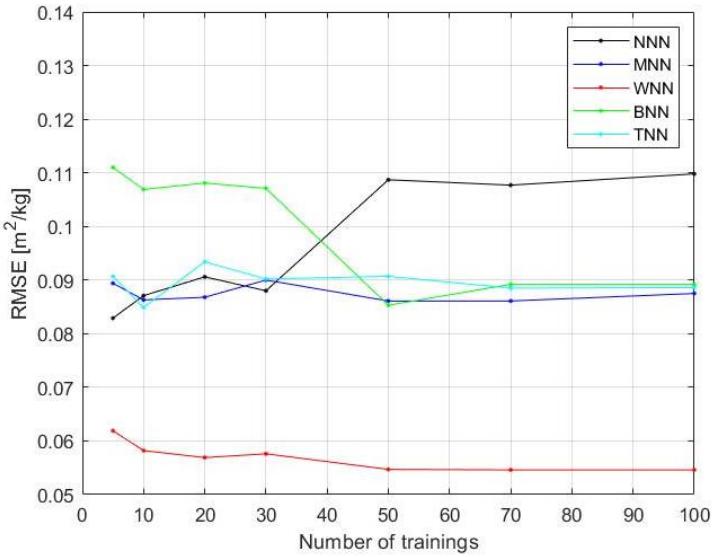


Figure 3.5 - RMSE for each Neural Network as a function of the number of trainings.

Propagation time

Another important aspect is to evaluate the accuracy of the prediction with respect to the propagation time. In fact, in real applications, especially after breakup events, information on space objects might not be available before a certain time. On the other hand, early data might not contain clear information for the learner as the orbital parameters do not exhibit a clear trend. With this regard, for each RSO of the testing set the median of the N_{tr} predictions is computed every day. These values will be referred to as $BC_{pred,day}$. For every $BC_{pred,day}$, the absolute value of the percentage error is computed as:

$$\Delta BC_{day} = \left| \frac{BC_{pred,day} - BC_{true}}{BC_{true}} \right| * 100\% \qquad (3.6)$$

hence having a distribution of ΔBC_{day} for each day. The mean of such distributions for the N_{obj} objects ($\mu_{\Delta BC,day}$) is computed and shown in Figure 3.6 for the five neural networks. It is evident that for all the neural networks, a significant reduction of the mean percentage error is achieved for a propagation time greater than 6 days, while for the BNN and TNN it is achieved after 3 days. This might be due to the fact that, for the considered ranges of orbital parameters and *AMR*, there is not a clear trend of

the orbital elements in the first 6 days since the effects of atmospheric drag on the mean orbital parameters are not large enough to provide observability for the ballistic coefficient. Moreover, for a propagation time greater than 6 days, it seems that the $\mu_{\Delta BC,day}$ is stabilised around a mean value. The WNN has the lowest $\mu_{\Delta BC,day}$ with respect to the other neural networks. Considering these results, it is interesting to evaluate the performance of the neural networks when data for a propagation time lower than a certain threshold are removed from the training set. To this purpose, thresholds of 5, 10, 15, 20 and 25 days have been defined and information at time lower than them have been removed. Figure 3.7 shows the RMSE with respect to the time interval considered for the propagation. It can be noticed that the RMSE reaches a minimum value for a time threshold of 5 or 10 days. This is in agreement with Figure 3.6, since information over time less than 6 days do not improve the learning process of the prediction model. If the threshold is set larger than 10 days, instead, the RMSE starts to increase. In fact, the observability of the drag effects does not improve after 10 days (as mentioned above). Hence, removing data before a threshold greater than 10 days, simply leads to a reduction of information and thus a less accurate BC estimation.

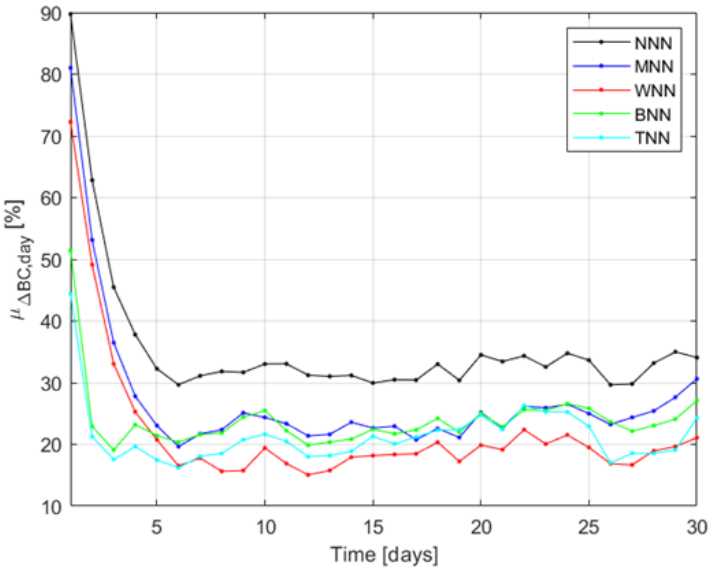


Figure 3.6 - Mean percentage error with respect to the propagation time.

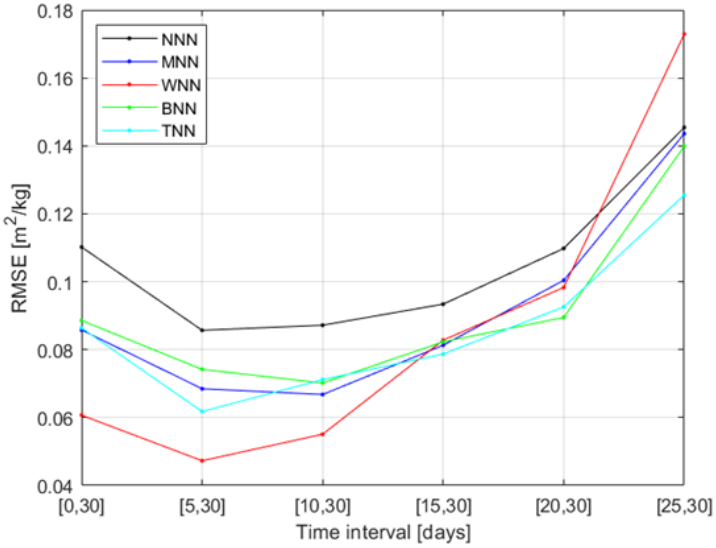


Figure 3.7 - RMSE vs. propagation time intervals.

Number of objects

The number of objects of the training set affects the performance of the prediction model. In fact, machine learning techniques, especially neural networks, can face two different issues: underfitting and overfitting [120]. In the former case, the number of objects is too low and the prediction model is incapable of capturing the variability of the data. On the other hand, overfitting can occur when the number of objects is too high and the neural network, at a certain time during the training period, does not improve its ability to make predictions, but rather learns too many details in the training data along with the noise inevitably contained. To analyse the impact that the number of objects in the training set has on the performance of the neural networks, such number has been progressively reduced following different criteria. A first criterium (*criterium 1*) has been to reduce $N_{obj,tr}$ by a factor of 2, 4 and 8 by randomly removing objects from the training set, thus obtaining data set of 825, 412 and 206 objects. For each data set, BC_{pred} values are computed as described in Section 3.2.1. Figure 3.8 shows the RMSE with respect to the number of objects. It can be noticed that, in all the cases, reducing the number of objects leads to an increase in the RMSE. This is more evident for the MNN and the WNN, while the NNN reaches a minimum

for 825 objects. This behaviour is coherent with the characteristics of the different neural networks, since the WNN is expected to work better with a high number of objects; conversely the NNN. The BNN and the TNN instead do not seem to be particularly affected by the number of objects.

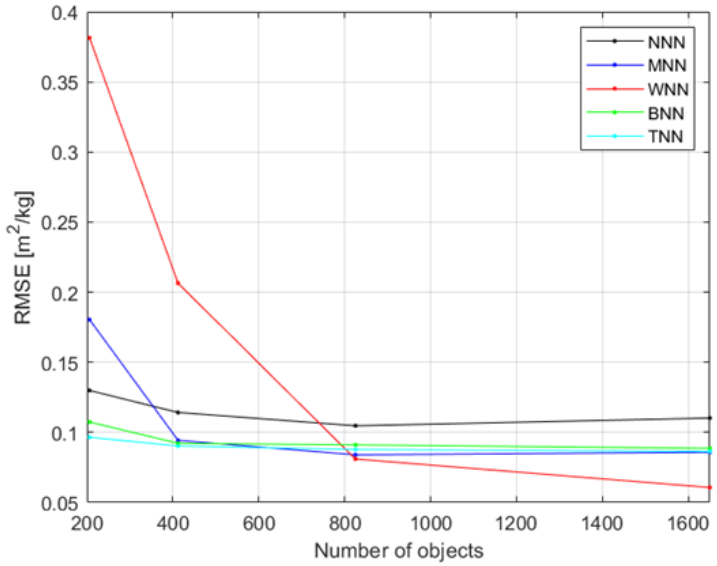


Figure 3.8 - RMSE of each Neural Network as a function of the number of objects, according to criterium 1.

A second criterium (*criterium 2*) has been to remove those objects that re-enter in the atmosphere (i.e., radius at perigee lower than 150 km) before a certain pre-defined time set equal to 10, 15, 20, 25 days. In fact, these objects exhibit an exponential variation of the semimajor axis during the re-entry phase and can thus affect the learning process. Applying this criterium, new data sets of 1642, 1563, 1478 and 1391 objects are obtained. For each data set, the RMSE has been computed as in Eq. 3.2. Figure 3.9 shows the RMSE with respect to the number of objects. It can be noticed that the NNN, MNN and WNN do not get an evident benefit from the removal of re-entered objects as the overall number of objects does not change significantly. On the other hand, BNN and TNN have a different trend, having a higher RMSE when re-entered objects are removed. This is probably due to the intrinsic structure of the neural network that is able to understand the different behaviour of re-entered objects.

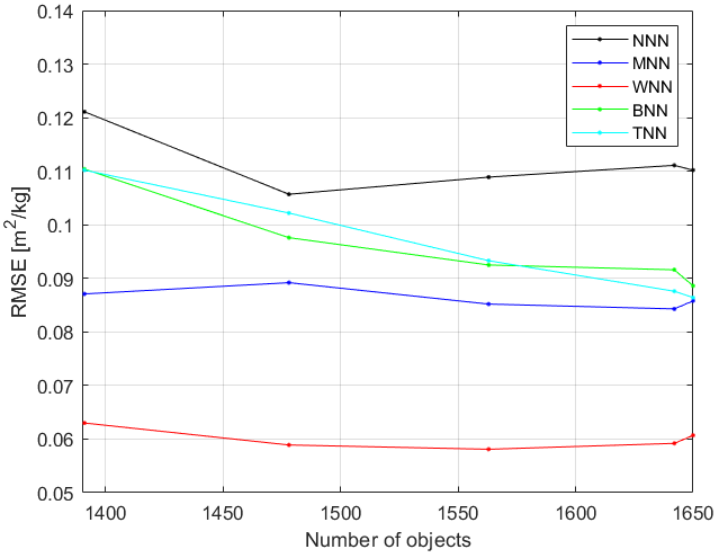


Figure 3.9 - RMSE of each Neural Network as a function of the number of objects, according to criterium 2.

Frequency of measurements

Another parameter that has been analysed is the frequency of the measurements available in the training and testing data sets. In the first case, it is important to understand whether a reduction of the sampling frequency in the training data set, that is beneficial for the computational cost, has an effect on the predictor performance when keeping the sampling frequency in the testing data set constant. To this purpose, the number of measurements per day has been decreased from one measurement per day (as in the initial training data set) to one measurement every 2, 3, 5, 10 days. Figure 3.10 shows the RMSE for the different sampling frequencies. It can be noticed that reducing the amount of information in the training set, increases considerably the RMSE since the predictor probably gets into underfitting. The second case is representative of an actual lack of information about the RSOs, due to the impossibility of ground-based sensors to properly track them (e.g., the RSO might not enter in the field of view of the sensor with a high frequency). Figure 3.11 shows the RMSE for the different sampling frequencies. It can be noticed that, also in this case, there is a reduction of the RMSE with the increase of the sampling frequency. However, such reduction is more modest than in the previous case and the value of the RMSE is pretty

constant for a frequency greater than one measurement every 2 days. This means that the predictor is quite robust with respect to the number of input measurements, provided that a sufficient quantity of data has been used for the training.

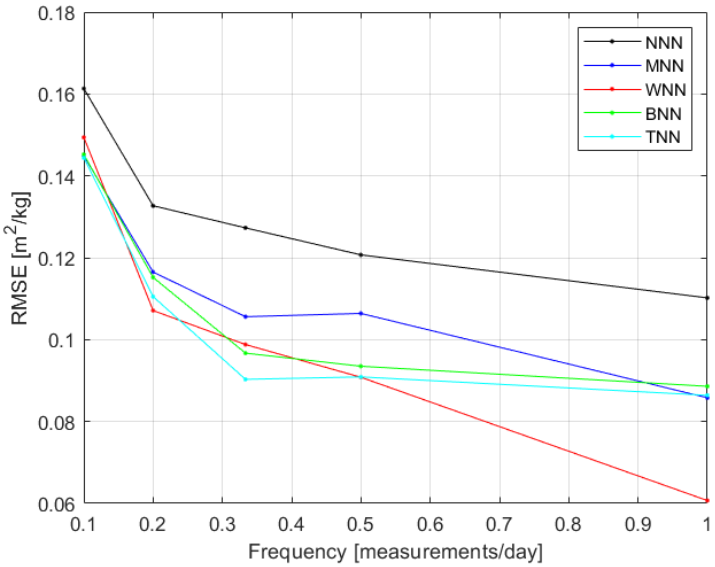


Figure 3.10 - RMSE of each Neural Network as a function of the sampling frequency in the training data set.

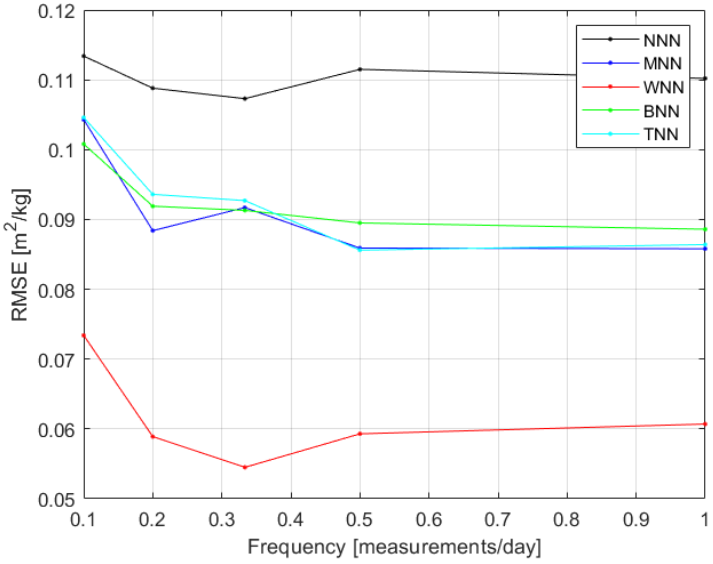


Figure 3.11 - RMSE of each Neural Network as a function of the sampling frequency in the testing data set.

Orbital decay prediction performance with modified training set

Considering the previous analyses, the training data set has been modified. In particular, keeping the number of objects equal to 1650 and the frequency of measurements equal to one measurement per day, information provided over time less than 5 days have been removed for the NNN, WNN and TNN, while information provided over time less than 10 days have been removed for the MNN and BNN. The number of trainings has been left equal to 10. For every object of the testing set, a value of BC_{pred} has been estimated with each neural network. Afterwards, the object has been propagated for 30 days and the mean semimajor axis thus obtained (a_{pred}) has been compared with the reference value (a_{true}). Defining Δa as:

$$\Delta a = \left| a_{pred} - a_{true} \right| \tag{3.7}$$

and assigning a threshold on Δa (indicated as Δa_{thr} and set equal to 2 km) to identify successful prediction of the *BC*, the performance of the modified training set can be assessed. Table 3.4 reports a "✓" symbol when Δa is lower than Δa_{thr} and an "✗" symbol if not. For each column, (i.e., for each object), it is reported the percentage of neural networks that provides a successful prediction. Similarly, for each row it is reported the percentage of positive results over the testing dataset obtained with a certain neural network. It can be noticed that the TNN is the neural network that exhibits the highest percentage of success, equal to 84%. Moreover, 70% of the objects can be properly predicted by all the neural networks. Figure 3.12 shows how the percentage of success increases when varying Δa_{thr} with values equal to 1 km, 3 km, 5 km and 10 km. It can be noticed how for Δa_{thr} equal to 1 km, the WNN has a higher percentage of success than the TNN, while for Δa_{thr} equal to 5 km all the neural networks have a percentage of success at least equal to 80%. Figure 3.13 shows the semimajor axis of objects with ID 15 and 21 as predicted by the different neural networks and the reference one. It can be seen how the predicted plots follow the trend of the reference plot. On the other hand, for some objects none of the neural networks is capable to properly estimate the *BC* and, hence, to predict the trend of the semimajor axis. These objects have mainly two characteristics: either they re-enter in the atmosphere or are characterized by a very low *BC*. So, in both cases the neural

networks are probably not able to capture the trend of the orbital elements as they exhibit either an exponential variation or a very low one. With this regard, Figure 3.14 (a) shows the semimajor axis of object with ID 7 (re-entered): even if the trend is followed properly for few days, the Δa increases noticeably in the last 4 days. Finally, figure 14 (b) shows an example (ID 11) in which only some neural networks have predicted the BC correctly (within a certain tolerance).

Table 3.4 - Neural Network assessment on training set objects. For each Neural Network and synthetic object, it is reported a "✓" symbol when Δa is lower than Δa_{thr} and an "×" symbol if not. The last row reports the percentage of how many NNs have provided a positive result. The last column reports the percentage of positive results of each NN for the complete testing data set.

	Object ID												
	1	2	3	4	5	6	7	8	9	10	11	12	13
NNN	✓	✓	×	✓	✓	✓	×	✓	✓	✓	×	✓	✓
MNN	✓	✓	×	✓	✓	×	×	✓	✓	✓	×	✓	✓
WNN	✓	✓	×	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
BNN	✓	✓	×	✓	✓	×	×	✓	✓	✓	✓	✓	✓
TNN	✓	✓	×	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
[%]	100	100	0	100	100	60	0	100	100	100	60	100	100

	14	15	16	17	18	19	20	21	22	23	24	25	26
NNN	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
MNN	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
WNN	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
BNN	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
TNN	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
[%]	100	100	40	100	100	100	100	100	0	100	100	100	100

	27	28	29	30	31	32	33	34	35	36	37	38	39
NNN	✓	✓	✓	×	✓	×	×	×	×	✓	×	×	✓
MNN	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	✓	✓
WNN	✓	✓	✓	×	✓	×	×	×	×	✓	✓	✓	✓
BNN	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	×	✓
TNN	✓	✓	✓	✓	✓	×	×	×	✓	✓	×	✓	✓
[%]	100	100	100	60	100	0	0	0	20	100	60	60	100

	40	41	42	43	44	45	46	47	48	49	50	[%]
NNN	×	✓	✓	✓	✓	×	×	✓	✓	✓	✓	70
MNN	×	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	76
WNN	×	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	80
BNN	×	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	76
TNN	×	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	84
[%]	0	100	100	100	100	0	100	100	100	100	100	

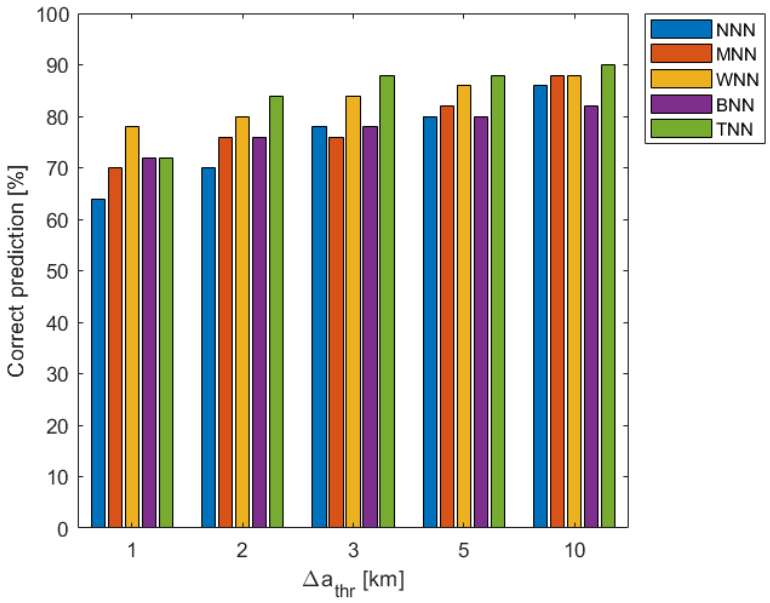
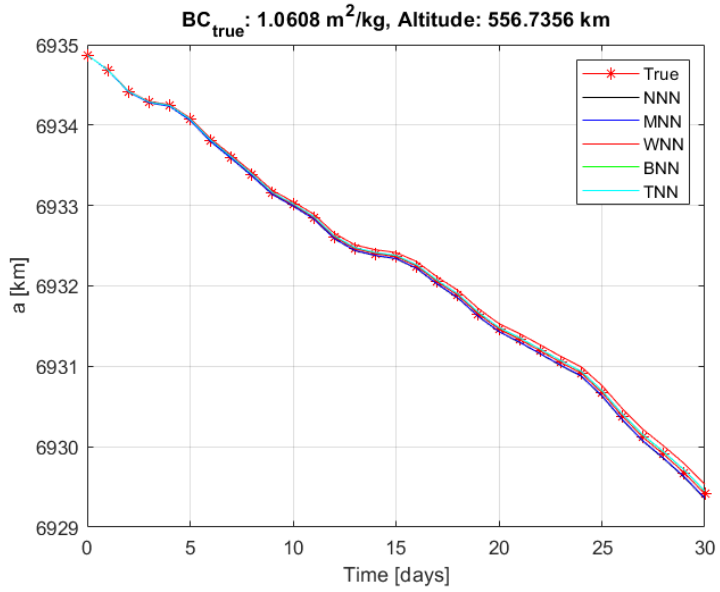
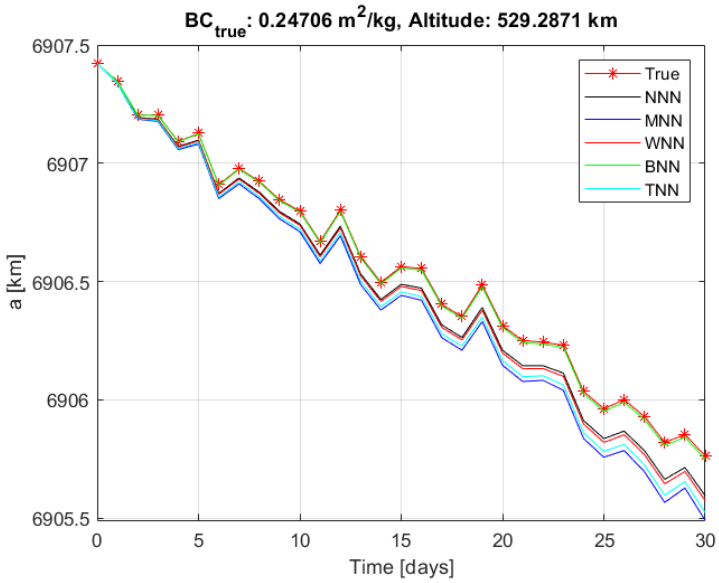


Figure 3.12 - Percentage of correct prediction for each Neural Network as a function of the Δa_{thr}.

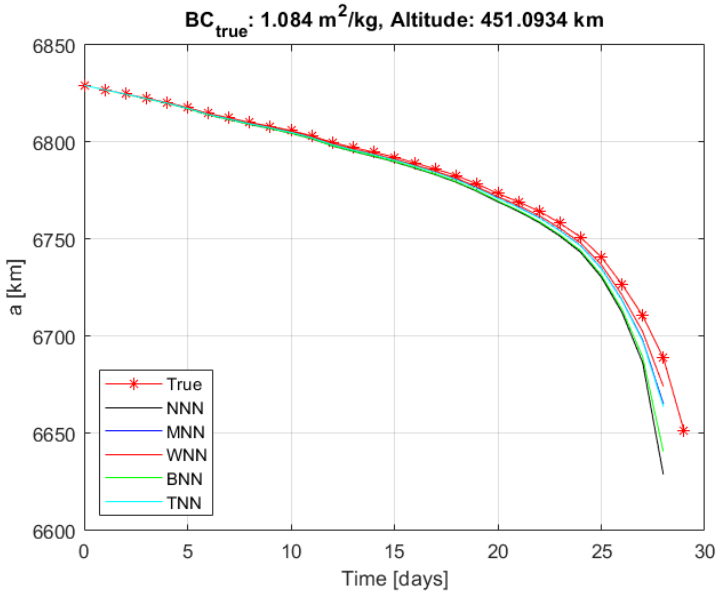


(a)

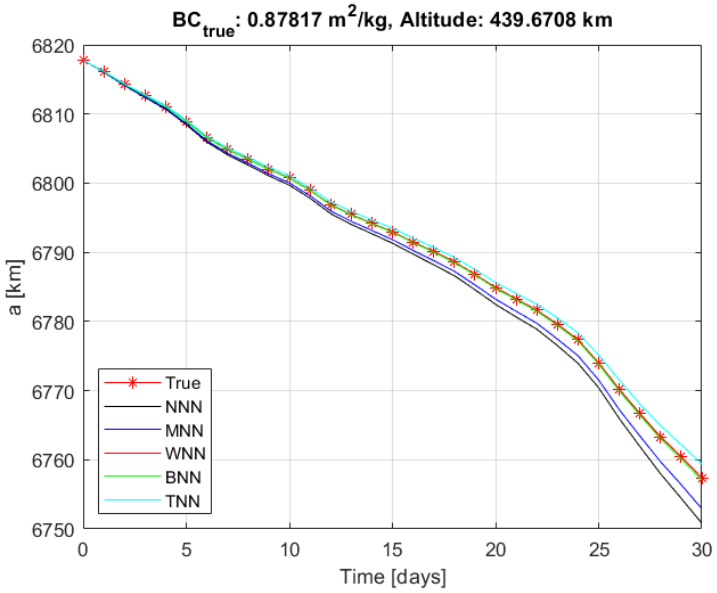


(b)

Figure 3.13 - Time evolution of the mean semimajor axis obtained with the *BC* predicted by each NN and compared to the true semimajor axis. Object #15 (a) and #21 (b).



(a)



(b)

Figure 3.14 - Time evolution of the mean semimajor axis obtained with the BC predicted by each NN and compared to the true semimajor axis. Object #7 (a) and #11 (b).

Considering the range of altitudes presented in this paper, it is important to highlight that an error of few kilometres on the semimajor axis can lead to a significant position error. Therefore, the information on Δa has been integrated with additional performance metrics. Since an error on the semimajor axis is associated to an error in the true anomaly, radial and along-track position errors (in a RSW reference frame, as defined in [113] have been considered.

In particular, due to the divergence of the position error at the altitudes of interest, this metric has been evaluated after 1 and 3 days of propagation. Table 3.5 and Table 3.6 report, for the WNN and the TNN (i.e., the two best NNs with respect to the Δa), the mean, the standard deviation and the median (over the testing data set) of the semimajor axis, true anomaly, radial and along-track errors. It is important to notice that the mean is, in most of the cases, one order of magnitude greater than the median. This is due to the presence of few objects in the testing set (i.e., objects with a *BC* smaller than 0.08 m^2/kg) for which an error in the prediction of the *BC* leads to a divergence of the position error. This is confirmed by the values of the standard deviation that are greater than 100% of the mean value.

Table 3.5 - Mean, standard deviation and median of the semimajor axis, true anomaly, radial and along-track position errors, evaluated after 1 and 3 days of propagation for the WNN.

	$\Delta a\ [km]$			$\Delta \nu\ [°]$		
N_{days}	μ	σ	m	μ	σ	m
1	0.040	0.092	0.006	0.022	0.049	0.003
3	0.129	0.332	0.015	0.206	0.497	0.023

	$\Delta r\ [km]$			$\Delta s\ [km]$		
N_{days}	μ	σ	m	μ	σ	m
1	0.036	0.083	0.006	2.878	6.393	0.456
3	0.315	0.934	0.018	26.261	59.117	4.034

Table 3.6 - Mean, standard deviation and median of the semimajor axis, true anomaly, radial and along-track position errors, evaluated after 1 and 3 days of propagation for the TNN.

	$\Delta a\ [km]$			$\Delta \nu\ [°]$		
N_{days}	μ	σ	m	μ	σ	m
1	0.054	0.150	0.007	0.024	0.049	0.005
3	0.183	0.540	0.016	0.235	0.635	0.026

	$\Delta r\ [km]$			$\Delta s\ [km]$		
N_{days}	μ	σ	m	μ	σ	m
1	0.050	0.137	0.007	3.807	9.997	0.467
3	0.668	2.347	0.015	34.679	91.942	4.137

In the results here presented, it is assumed that the initial state of the RSO is error free. However, to address the sensitivity of the method with respect to initialization and orbit determination errors, the following analysis has been carried out. A synthetic object from the testing data set has been first selected. The initial orbital elements of the object are reported in Table 3.7 and refer to the 1st of January 2021 at 00:00:00.000 UTC. The true *BC* is 0.699 m^2/kg. Then, the object has been propagated with the true *BC* and the orbital elements have been sampled as described in Section 3.2.1. An uncertainty in the knowledge of the position and the velocity for each sampled measurement has been assumed to simulate a measurement error (thus leading to an estimated state $\boldsymbol{X}_{est}$). This uncertainty has been modelled as a white Gaussian noise on the objects position and velocity in the RSW reference frame, with a standard deviation in the radial (σ_r), along-track (σ_s) and cross-track (σ_w) direction equal to 0.02 km, 0.2 km and 0.2 km for the position and 0.02 m/s on each component of the velocity respectively. These values have been selected as uncertainties associated to the orbit determination based on measurements updates. The initial measurement thus obtained ($\boldsymbol{X}_{est,in}$) has been used as initial state for the false propagation (i.e., with *BC* = 1 m^2/kg). Afterwards, the consistency error has been computed using the sampled estimated measurements $\boldsymbol{X}_{est}$ and the sampled data from the false propagation $\boldsymbol{X}_{BC1}$. Then, a *BC* has been predicted by each Neural Network using the same settings as the analysis on the testing data set. Figure 3.15 shows how the measurement errors are included in the testing architecture. A set of 100 simulations has been carried out performing a statistical analysis in the achieved results in terms of the percentage error on the estimated *BC*:

$$Err_{BC} = \left| \frac{BC_{pred} - BC_{true}}{BC_{true}} \right| 100\% \tag{3.8}$$

Figure 3.16 shows the boxplots of the Err_{BC} for each Neural Network. On each box, the central mark (i.e., the red line) indicates the median of the statistics parameter (i.e., Err_{BC}), the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The difference between the 75th and 25th percentiles (i.e., the height of the box) is called interquartile range (IQR). The whiskers extend to a value equal to 1.5 IQR above the 75th percentile and 1.5 IQR below the 25th percentile.

Table 3.7 - Initial orbital elements and true Ballistic Coefficient of the Resident Space Object with ID 5.

ID	a $[km]$	e $[-]$	i [°]	Ω [°]	ω [°]	ν [°]	BC [m^2/kg]
5	6904.61	1.28e-3	43.25	176.71	176.00	126.26	0.699

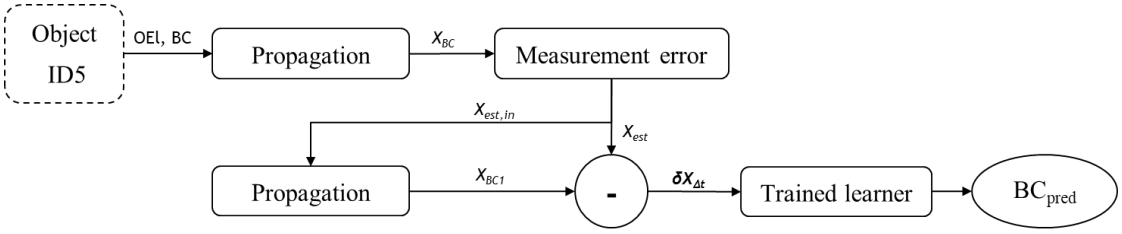


Figure 3.15 - Architecture of the method when a measurement error is considered.

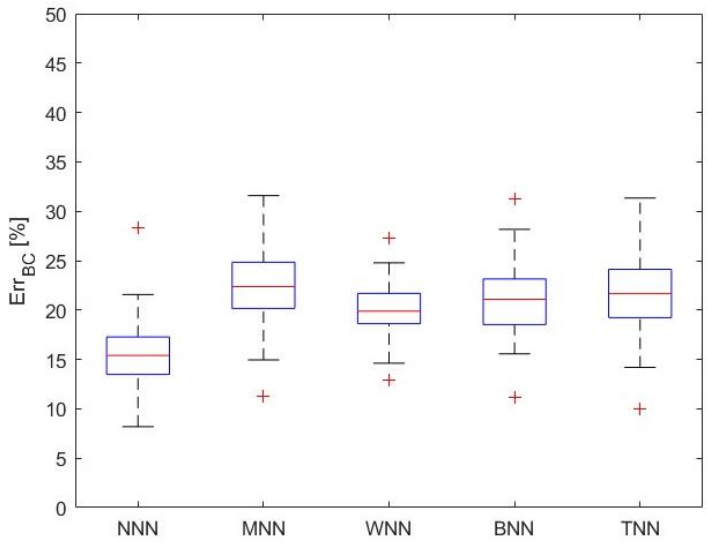


Figure 3.16 – Absolute value of the percentage error of the predicted *BC* for object with ID 5 of the testing data set. The boxplots refer to 100 different initial conditions. The red line represents the median of the error.

An important assumption in this section is that the truth and the model atmosphere are the same. In fact, the scope of these analyses is to assess the effects of the features set and the sensitivity with respect to the most relevant parameters (i.e., the number of objects, the propagation time, the frequency of measurement) on the method performance. Therefore, this assumption allows isolating such effects from the ones

due to a different atmospheric model. However, different density models can lead to different values of the atmospheric density, ρ_{atm}. This means that a certain variability can be associated to the value of ρ_{atm} predicted by an assumed atmospheric model. In particular, a maximum 100 % variation of ρ_{atm} can be considered [121]. This variation can be a good estimation of the atmospheric density uncertainty involved, and so of the drag acceleration variation one can expect when dealing with objects characterized by a stable history of the ballistic coefficient. The impact of such atmospheric mismodeling on the performance of the method could be evaluated either propagating the testing set objects using atmospheric models that are different to the one used in the training phase or, similarly, introducing an uncertainty on the atmospheric density.

Upgrades to include space weather effects

The atmospheric density ρ_{atm} plays an important role in the perturbing acceleration term associated to the atmospheric drag, thus affecting the time evolution of the semimajor axis, and hence the observability of the *BC*. The value of ρ_{atm} strongly depends on the space weather [122] whose variability is affected by two major drivers: the solar irradiance and the geomagnetic activity. Such variability is reflected by the indexes that describe the space weather: the planetary range index (K_p), the planetary equivalent amplitude (A_p), the observed 10.7-cm solar radio flux ($F_{10.7}$). Errors in the forecast of these indexes can lead to position errors (mainly in the along-track directions) of few tens of meters within one day [123], that diverge in time [124]. Different models (e.g., Brownian motion, White noise) have been developed to consider the stochastic nature of atmospheric density variability [124]. Reproducing these models is out of the scope of this work.

However, the effects of atmospheric density variability can be considered adding the geomagnetic index used in the adopted atmospheric model (i.e., the K_p in the Jacchia-Bowman) as a further feature of the Neural Networks. To consider the K_p effects in the training phase, the 30 days-propagation described in Section 3.2.1 has been performed with two initial epochs: 1st January 2021 and 10th April 2021

corresponding to an average value, over 30 days, of K_p equal to 0.94 and 1.58 respectively. The two sets of data thus obtained have been merged. A further binary feature has been introduced and set equal to 1 for the data corresponding to January (i.e., low K_p) and to 2 for the data corresponding to April (i.e., high K_p).

To assess the performance of the ML-technique, when the solar activity effects are considered in the training set, the following analysis has been carried out. The space objects of the testing set have been propagated for 30 days with initial epoch set to 10th April 2021. Afterwards, the ballistic coefficient of these objects has been estimated using three different training data sets:

1. Training set with initial epoch at 1st January 2021, without the K_p feature (*TS1*)

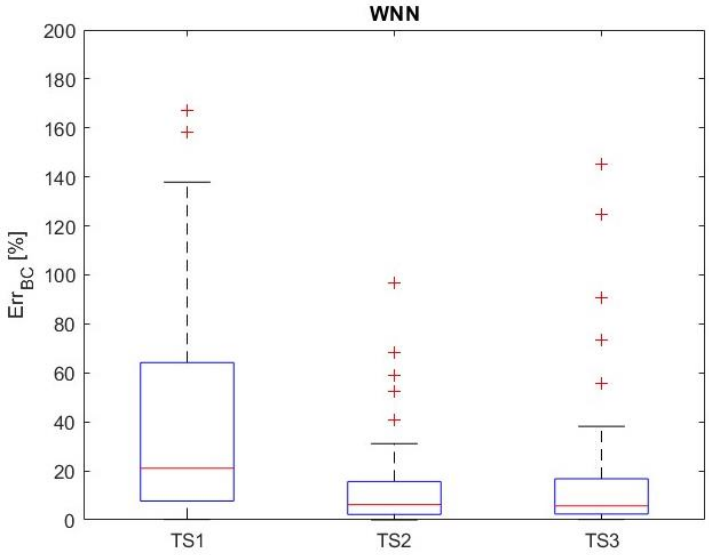
2. Training set with initial epoch at 1st January 2021 merged with the training set with initial epoch set to 10th April 2021, with the K_p feature (*TS2*)

3. Training set with initial epoch at 10th April 2021, without the K_p feature (*TS3*)

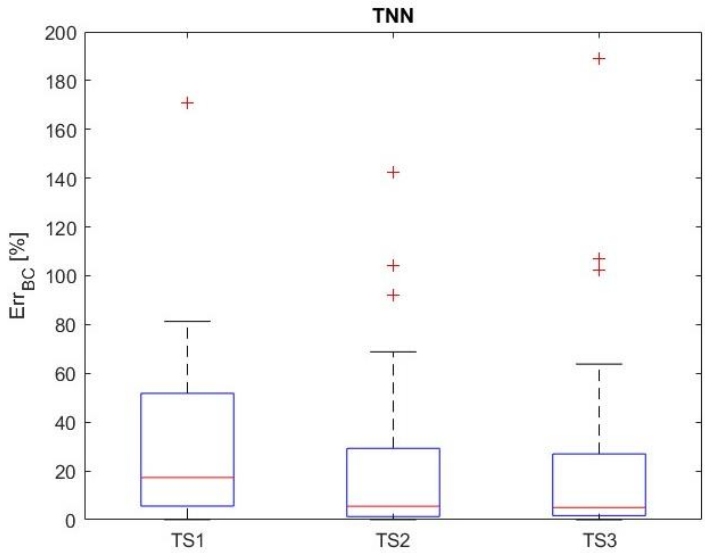
For each case, and for each NN, the Err_{BC} can be computed as in Eq. 3.8. Table 3.8 reports the mean and the standard deviation of the Err_{BC} for each neural network, whereas Figure 3.17 represents the boxplots of the Err_{BC} for the WNN (a) and the TNN (b). It is evident that the solar activity indexes must be included in the training set when different periods of the years are considered. In fact, the performance improves significantly using TS2 and TS3 instead of TS1. In particular, using TS2 and TS3 leads to very similar performance. This means that the binary feature of the K_p used in TS2 can properly capture the difference between the data coming from TS1 and TS3.

Table 3.8 - Mean and standard deviation of the percentage error for three different training sets.

	TS1		TS2		TS3	
NN	μ [%]	σ [%]	μ [%]	σ [%]	μ [%]	σ [%]
NNN	34.55	35.49	30.82	51.57	28.95	45.43
MNN	48.68	70.76	23.23	34.96	20.13	30.23
WNN	64.31	118.0	13.24	19.34	17.31	30.36
BNN	41.42	49.99	20.80	32.80	20.33	33.25
TNN	30.92	32.85	22.92	39.31	20.69	35.03



(a)



(b)

Figure 3.17 - Boxplot of the mean percentage error for the WNN (a) and the TNN (b), when the K_p is introduced in the training set.

3.2.3 Results on Resident Space Objects

TLE data from real space objects are used to obtain additional test cases to assess the proposed approach performance. A first set of objects is composed of 10 debris with initial orbital elements as reported in Table 3.9. These values are extracted from TLEs taken from SpaceTrack [114] and refer to the first available TLE in January 2021, i.e., to the epoch reported in table 10 in UTC format. Then, all the available TLEs within a time interval of 30 days have been considered for the BC estimation. In the last column of Table 3.10, the type of debris is indicated, i.e., whether it comes from a rocket or a payload breakup. For each RSO, the ballistic coefficient is estimated with every neural network (see Table 3.11). The object is then propagated with these predicted values and the semimajor axis compared with the one derived from the TLEs. The propagation is performed with the numerical propagator and orbital perturbations described in Section 3.2.1. For the time interval of interest (1-31 January 2021), the K_p index is in the range [0, 4.70]. The maximum Δa (Δa_{max}) is computed for each neural network and shown in Figure 3.18. It can be noticed that for all the debris,

except for ID 46202, there is at least one neural network that leads to a Δa_{max} lower than 1 km and the WNN always leads to Δa_{max} lower than 2 km. It must be highlighted that there are five main differences between the numerical results obtained using the testing dataset and those obtained using as input TLE data from debris in orbit. First, the atmospheric model and the C_d value used in the testing set are known and the error in the propagation can be associated only to an error in the *AMR* estimation. In the real test cases, instead, the atmospheric model and the C_d are unknown and the Δa_{max} can depend on all the three parameters. This means that uncertainties related to an atmospheric density or drag coefficient mismodeling are included in the *BC* prediction. Second, the model used to convert the orbital elements from osculating to mean (i.e., Brouwer theory) is not perfectly consistent with the SGP4 model adopted within TLE generation. Third, TLEs result from orbital determination and are characterised by a given accuracy, which is not considered in the training process. Fourth, when the algorithm is applied to TLEs, there can be errors associated to the interpolation. Finally, none of the considered debris re-enter within 30 days. This latter point has probably had a positive impact on the overall performance. For object ID 46202, none of the neural networks has predicted correctly the BC. This might depend on the difference between the atmospheric model used in the training data set and the one used for the TLEs determination. Figure 3.19 and Figure 3.20 show the semimajor axis obtained using the predicted *BC* with respect to the one obtained from the TLEs.

Table 3.9 - Initial orbital elements of the Resident Space Objects.

ID	a $[km]$	e $[-]$	i [°]	Ω [°]	ω [°]	ν [°]
28291	6854.04	3.16e-3	75.61	291.55	91.66	268.30
32075	6863.13	8.38e-4	98.25	153.09	60.24	299.87
33909	6898.13	1.97e-3	74.16	94.54	68.23	291.78
34333	6831.85	1.63e-4	74.12	63.61	201.85	158.10
34448	6878.59	1.44e-4	73.94	234.74	80.82	279.25
34631	6853.67	1.52e-4	74.12	92.52	355.73	4.27
40326	6900.82	2.06e-3	86.47	41.38	82.09	277.82
46185	6938.45	3.98e-3	97.94	123.67	140.04	220.03

46200	6939.73	3.67e-3	97.92	122.76	157.38	202.68
46202	6957.07	1.54e-3	97.93	122.27	211.53	148.54

Table 3.10 - Initial epoch in UTC format and type of debris.

ID	Initial Epoch (UTC) [YYYY-MM-DDThh:mm:ss.sss]	Type of debris
28291	2021-01-01T05:19:42.576	Rocket
32075	2021-01-01T20:52:03.930	Payload
33909	2021-01-01T15:47:06.993	Payload
34333	2021-01-01T13:29:54.161	Payload
34448	2021-01-01T16:51:12.713	Payload
34631	2021-01-04T07:46:25.796	Payload
40326	2021-01-01T20:36:42.959	Payload
46185	2021-01-01T19:23:57.913	Rocket
46200	2021-01-01T18:38:11.130	Rocket
46202	2021-01-01T20:09:13.961	Rocket

Table 3.11 - BC predicted by each Neural Networks for each real Resident Space Object.

	Predicted BC [m^2/kg]									
	28291	32075	34333	34448	34631	33909	40326	46185	46200	46202
NNN	0.292	0.342	0.339	0.449	0.288	1.076	1.205	1.276	1.363	1.453
MNN	0.501	0.492	0.373	0.565	0.351	1.021	1.138	1.466	1.396	1.507
WNN	0.428	0.465	0.308	0.550	0.428	0.996	1.124	1.660	2.027	1.719
BNN	0.280	0.463	0.377	0.543	0.332	1.139	1.193	1.372	1.651	1.709
TNN	0.359	0.473	0.362	0.436	0.330	1.260	1.301	1.401	1.464	1.677

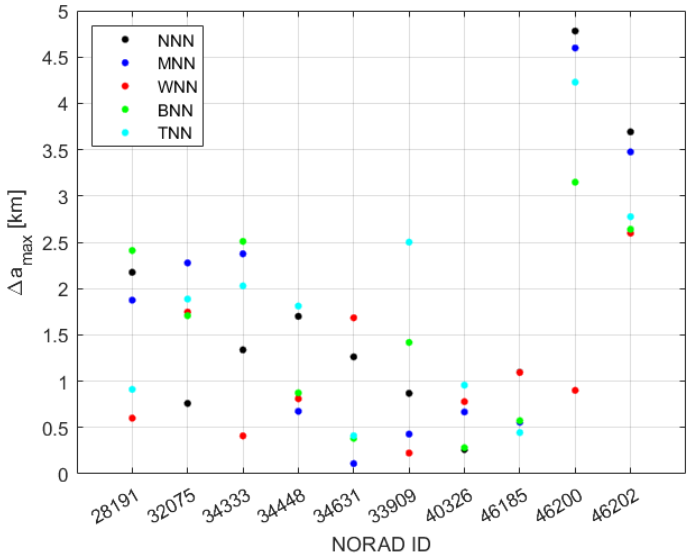
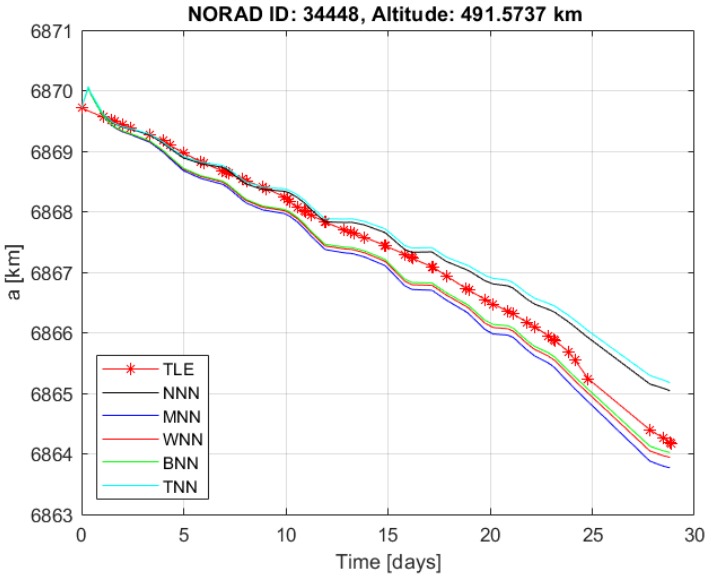
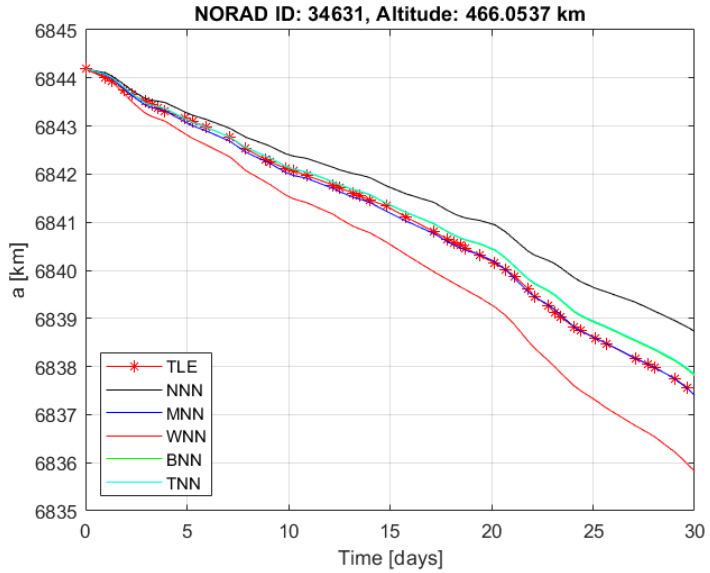


Figure 3.18 - Maximum Δa, evaluated after 30 days of propagation, for real RSOs.

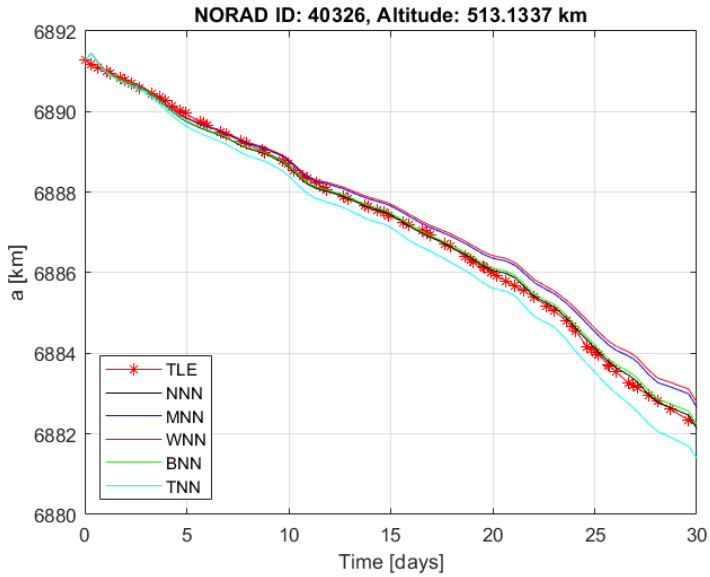


(a)

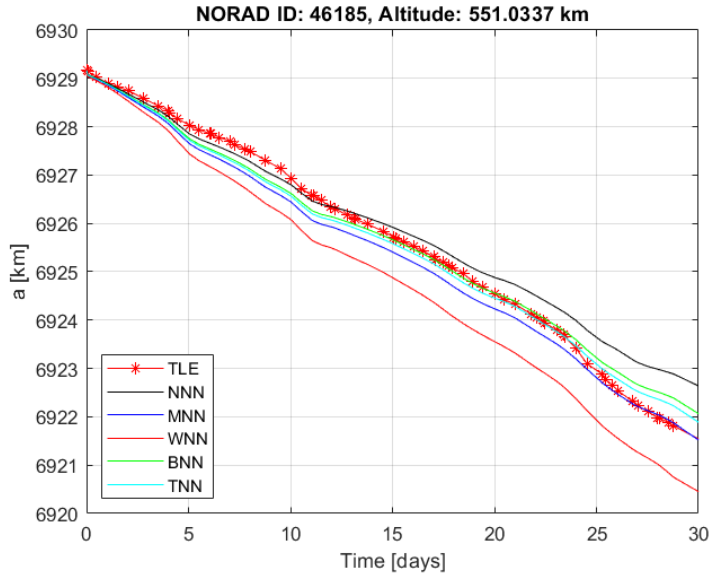


(b)

Figure 3.19 - Time evolution of the mean semimajor axis obtained with the *BC* predicted by each NN and compared to the semimajor axis extracted from TLEs. Objects with NORAD ID 34448 (a) and 34631 (b).



(a)



(b)

Figure 3.20 - Time evolution of the mean semimajor axis obtained with the BC predicted by each NN and compared to the semimajor axis extracted from TLEs. Objects with NORAD ID 40326 (a) and 46185 (b).

The algorithm can be applied also in cases in which a priori information about the space object characteristics is available, so that the *BC* estimate can be linked with the attitude behaviour. As an example in this framework, a second set of objects is composed of two satellites of Lemur - 2 constellation [125], in particular Lemur -2 Angela (NORAD ID: 42752) [126] and Lemur -2 Zupanski (NORAD ID: 43695) [127]. Each satellite is a 3U-Cubesat of launch mass equal to 4 kg and of size 10 cm x 10 cm x 34.5 cm [128], featuring two solar panels each with two (or three) segments of size 10 cm x 34.5 cm. For each satellite, the minimum and maximum ballistic coefficient can be computed. The former, corresponding to the configuration with the solar panels parallel to the velocity vector, is equal to BC_{min} = 0.030 m^2/kg (both for the 2nd and 3rd generation of satellites). The latter, corresponding to the configuration with the solar panels perpendicular to the velocity vector, is equal to BC_{max} = 0.103 m^2/kg for the 2nd generation of satellites and BC_{max} = 0.141 m^2/kg for the 3rd generation of satellites. The initial orbital elements and the relative reference epochs (expressed in UTC format) are reported in Table 3.12. These values refer to the first

available TLE in January 2021. Similar to the previous case, all the available TLEs within the time interval 1st -31st of January 2021 have been considered. For each satellite, the ballistic coefficient is estimated with every neural network (see Table 3.13). The satellite is then propagated with these predicted values and the semimajor axis compared with the one derived from the TLEs. The maximum Δa, Δa_{max}, is computed for each neural network and shown in Figure 3.21. It can be noticed that all the neural networks lead to a Δa_{max} lower than 2 km and the MNN and WNN lead to Δa_{max} lower than 1 km. Moreover, the predicted *BC* (Table 3.13) are consistent with the ones estimated from the physical dimensions. In particular the WNN, that minimizes the Δa_{max}, predicts a *BC* that is very close to BC_{min}. This might suggest that the satellite has an attitude similar to the one corresponding to BC_{min}. Figure 3.22 shows the semimajor axis obtained using the predicted *BC* with respect to the one obtained from the TLEs. It must be highlighted that, also in this case, as in the application of the algorithm to space debris, the actual atmospheric model is unknown. Therefore, the uncertainty related to the atmospheric density mismodeling is included in the determination of the *BC*.

Table 3.12 - Initial orbital elements and initial epoch in UCT format of Lemur-2 constellation satellites. First row refers to NORAD ID 42752. Second row refers to NORAD ID 43695.

a [km]	e [−]	i [°]	Ω [°]	ω [°]	ν [°]	Initial Epoch (UTC) [YYYY-MM-DDThh:mm:ss.sss]
6852.69	1.29e-3	51.56	300.30	5.60	354.33	2021-01-01T04:21:33.832
6884.44	1.32e-3	84.94	306.36	352.64	7.29	2021-01-01T09:12:15.267

Table 3.13 - *BC* predicted by each Neural Networks for Lemur-2 constellation satellites. Reference values of minimum and maximum *BC*, based on geometrical and attitude information, are provided.

	Predicted *BC* [m²/kg]		Reference Values		
	42752	4369	$\boldsymbol{BC_{min}}$ [m²/kg]	$\boldsymbol{BC_{max}}$ [m²/kg] (2nd generation)	$\boldsymbol{BC_{max}}$ [m²/kg] (3rd generation)
NNN	0.113	0.038			
MNN	0.051	0.030			
WNN	0.049	0.050			
BNN	0.069	0.067	0.030	0.103	0.141
TNN	0.082	0.083			

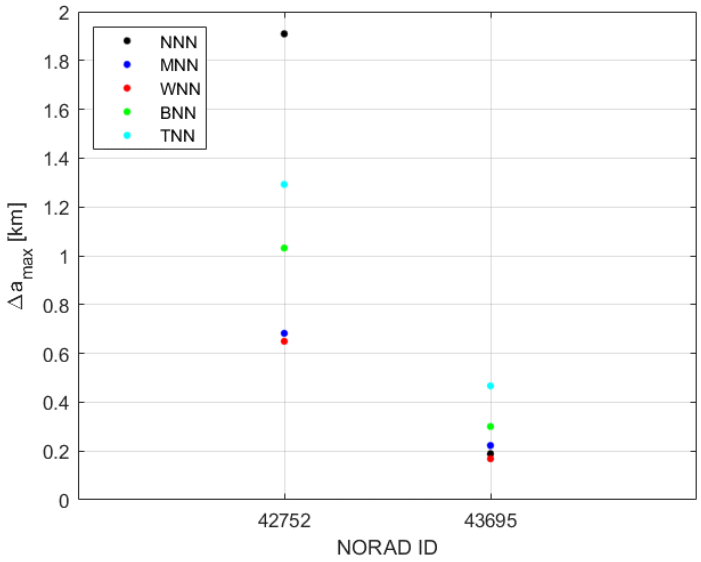
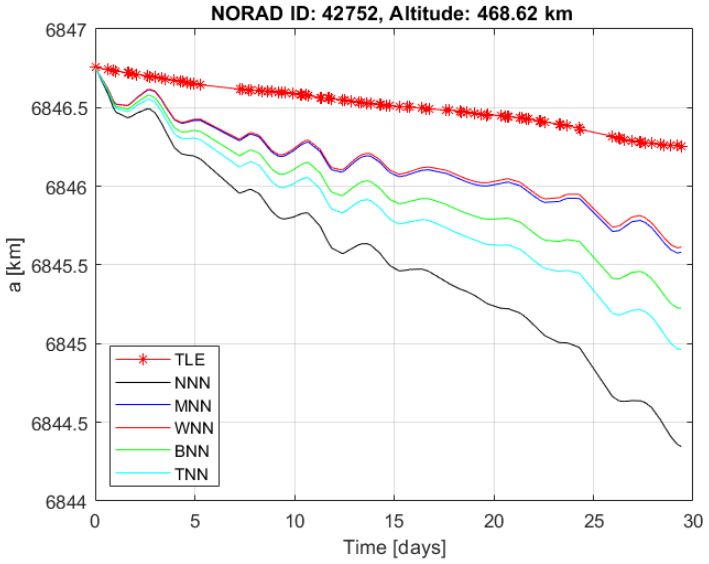
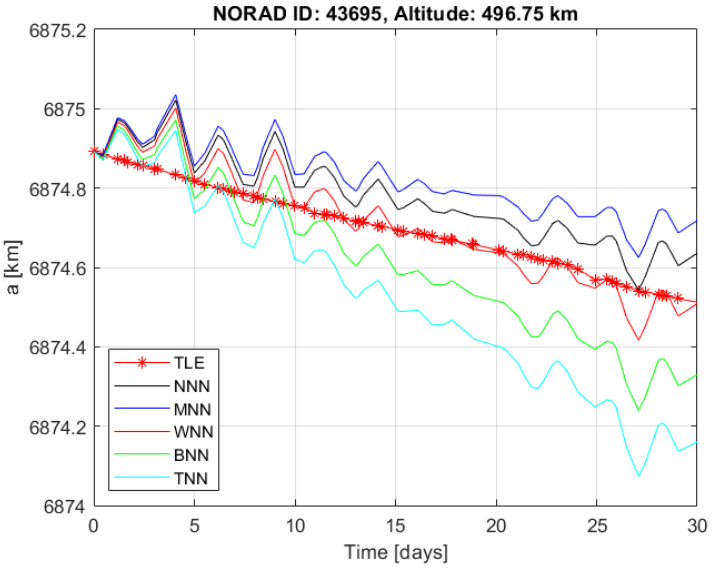


Figure 3.21 - Maximum Δa , evaluated after 30 days of propagation, for Lemur-2 constellation satellites.



(a)



(b)

Figure 3.22 - Time evolution of the mean semimajor axis obtained with the *BC* predicted by each NN and compared to the semimajor axis extracted from TLEs. Object with NORAD ID 42752 (a) and 43695 (b).

3.2.4 Comparison with a Semi-analytical method

In this section, the ML-based approach is compared with a traditional physics-based, semi-analytical approach in terms of accuracy and computational cost, as well as in terms of sensitivity with respect to the propagation time and the frequency of data.

Semi-analytical method description

The semi-analytical method (SAM) implemented in this section consists of two steps. A first guess of the area-to-mass ratio, hereinafter called *a-priori AMR*, is obtained by the numerical integration of the second term of Eq. 3.9, where $a(t)$ is the semi-major axis at date t, μ the Earth's gravitational constant, A_{drag} the cross-sectional area, m the object mass, ρ the atmospheric density, C_d the drag coefficient (set equal to 2.2) and V the norm of the spacecraft velocity, as a non-rotating atmosphere is considered. It is initially assumed that the mean evolution of the semi-major axis with time is exclusively due to the drag perturbing acceleration.

$$\frac{1}{a(t_0)} - \frac{1}{a(t)} = -\frac{1}{\mu}\frac{A_{drag}}{m}\int_{t_0}^{t} \rho C_d V^3 dt \tag{3.9}$$

To refine this initial estimation of the area-to-mass ratio, considering both drag and solar radiation pressure, the time derivatives of the orbital parameters, provided by the numerical propagator described in Section 3.2.1, are used. The contributions to these time derivatives (characterizing the temporal evolution of semi-major axis and eccentricity) due to the drag and solar radiation pressure forces can be computed separately using the Gauss V.o.P. equations [113], [129].

$$\begin{aligned} \left.\frac{da}{dt}\right|_{prop} &= \left.\frac{da}{dt}\right|_{drag} + \left.\frac{da}{dt}\right|_{SRP} \\ \left.\frac{de}{dt}\right|_{prop} &= \left.\frac{de}{dt}\right|_{drag} + \left.\frac{de}{dt}\right|_{SRP} \end{aligned} \tag{3.10}$$

Afterwards, the time derivatives of the same orbital parameters are computed, considering their time evolution as extracted from the TLEs. Eq. 3.11 give the relations

between the time derivatives provided by the propagator and those observed from the TLE catalogue. Since the time derivatives of the orbital parameters have been computed using the a-priori *AMR*, a multiplicative coefficient must be added to them so as to correctly fit the actual evolution observed from the TLEs:

$$\left.\frac{da}{dt}\right|_{TLEs} = K_1 \left.\frac{da}{dt}\right|_{drag} + K_2 \left.\frac{da}{dt}\right|_{SRP}$$
$$\left.\frac{de}{dt}\right|_{TLEs} = K_1 \left.\frac{de}{dt}\right|_{drag} + K_2 \left.\frac{de}{dt}\right|_{SRP} \quad (3.11)$$

where $\left.\frac{da}{dt}\right|_{TLEs}$ and $\left.\frac{de}{dt}\right|_{TLEs}$ represent the time derivatives of the semi-major axis and the eccentricity obtained by the TLEs in input, $\left.\frac{da}{dt}\right|_{drag}$ and $\left.\frac{de}{dt}\right|_{drag}$ the time derivatives due to the aerodynamic drag obtained by the propagator, and $\left.\frac{da}{dt}\right|_{SRP}$ and $\left.\frac{de}{dt}\right|_{SRP}$ the ones due to the SRP obtained by the propagator.

For each one of the considered TLEs, a pair of K_1 and K_2 is computed using the logic below:

- the mean of the time derivative of the semi-major axis cumulated from the epoch of the first TLE up to the current TLE is computed;
- the mean of the contributions to the semi-major axis (drag and SRP) cumulated from the initial epoch up to the current epoch is computed (based on propagated values);
- the same process is applied for the eccentricity;
- Eq. 3.11 can therefore be solved.

Finally, the median of this set of K_1 and K_2 values is computed to obtain the final multiplicative coefficient. Hence, the values of the dynamic parameters that will be used as initial guess for the orbit determination process, called *refined* area-to-mass ratio, are obtained as follows:

$$S_{drag}/m\big|_{ref} = K_1 \cdot S_{drag}/m\big|_{a\ priori}$$
$$S_{SRP}/m\big|_{ref} = K_2 \cdot S_{SRP}/m\big|_{a\ priori} \tag{3.12}$$

Performance assessment and comparison with feedforward neural networks

The comparison between the ML-based and the semi-analytical approaches has been evaluated using a testing dataset (i.e., 50 synthetic objects generated as described in Section 3.2.1, having as initial epoch 1st January 2021). Also for the SAM, mean orbital elements have been used for the time derivatives computation.

For each object of the testing set, a *BC* is estimated using the Neural Networks and the semi-analytical method, namely BC_{pred}. Defining as BC_{true} the actual ballistic coefficient of the object, it is possible to compute the absolute value of the percentage error Err_{BC} as in Eq. 3.8. In this way a set of 50 Err_{BC} for each NN and the SAM is obtained. Figure 3.23 shows the boxplots of the Err_{BC} for the five NN and the SAM. Table 3.14 reports the median (m) and the IQR of the Err_{BC}. It can be noticed that the WNN has the lowest median and all the NNs have a median lower than the SAM. Moreover, the WNN and the SAM exhibit a much lower IQR compared to the other NNs, hence a lower variance of the estimated values. As regards the computational cost, Table 3.15 reports the time necessary to estimate the BC of one object of the testing set. The NNs and the SAM have a comparable computational time, mainly due to the numerical propagation.

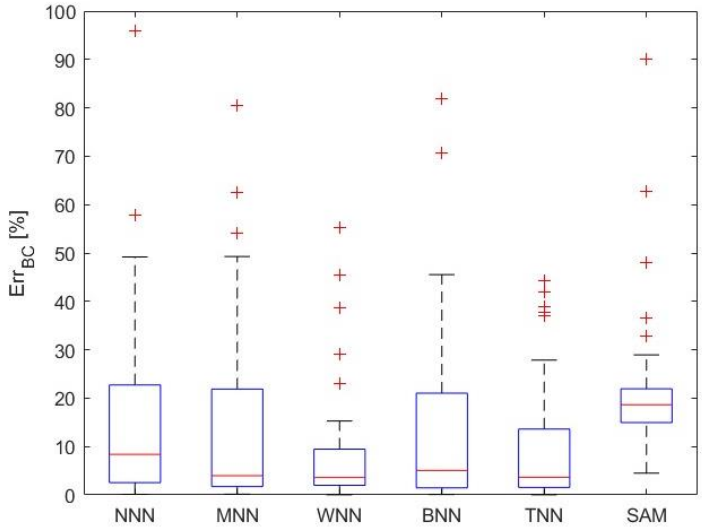


Figure 3.23 - Boxplot of the absolute value of the percentage error in simulated test cases.

Table 3.14 - Median (m) and IQR of the absolute value of the percentage error in simulated test cases.

	NNN	MNN	WNN	BNN	TNN	SAM
m	8.4	4.0	3.6	5.0	3.7	18.6
IQR	20.2	20.2	7.5	19.6	12.1	7.0

Table 3.15 - Computational cost for estimating the *BC*.

	Propagation time [s]	Estimation time [s]	Total time [s]
NNN		2.4	242.4
MNN		2.4	242.4
WNN	240	24.0	264
BNN		6.0	246
TNN		9.6	249.6
SAM	240	-	240

To evaluate the sensitivity analysis with respect to the propagation time, a ΔBC_{day} is computed also with the daily *BC* predicted by the SAM, according to Eq. 3.6, hence having a distribution of ΔBC_{day} for each day. The mean of such distributions for the $N_{obj,te}$ objects ($\mu_{\Delta BC,day}$) is computed and shown in Figure 3.24 for the five neural networks and the SAM. It is evident that the SAM, similarly to the NNs, exhibits a significant reduction of the mean percentage error for a propagation time greater than 6 days. This might be due to the fact that, for the considered ranges of orbital parameters and AMR, there is not a clear trend of the orbital elements in the first 6 days since the effects of atmospheric drag on the mean orbital parameters are not large enough to provide observability for the ballistic coefficient. Moreover, for a propagation time greater than 6 days, the $\mu_{\Delta BC,day}$ is stabilised around a mean value. After 6 days, the SAM has a mean percentage error evolution comparable with the MNN, the BNN and the TNN.

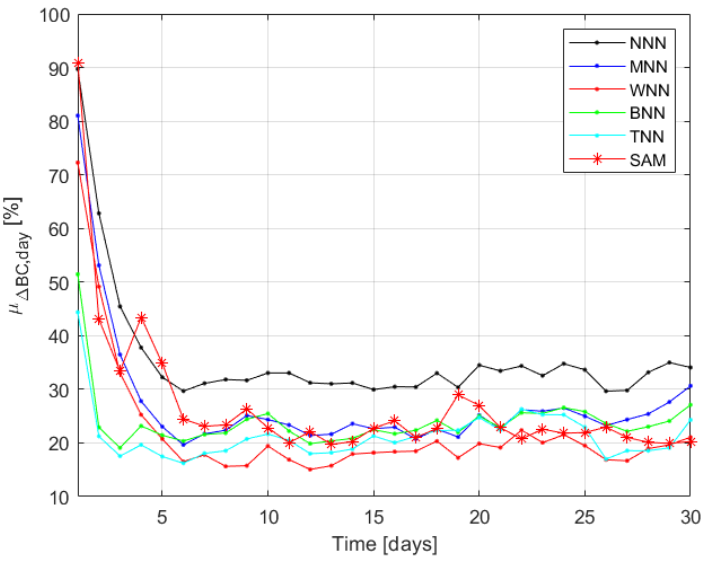


Figure 3.24 - Mean percentage error with respect to the propagation time.

Finally, to analyse the sensitivity with respect to the frequency of the measurements in the testing set, the number of measurements per day has been decreased from one measurement per day (as in the initial testing data set) to one measurement every 2, 3, 5, 10 days. Figure 3.25 shows the RMSE for the different sampling frequencies, for the NNs and the SAM.

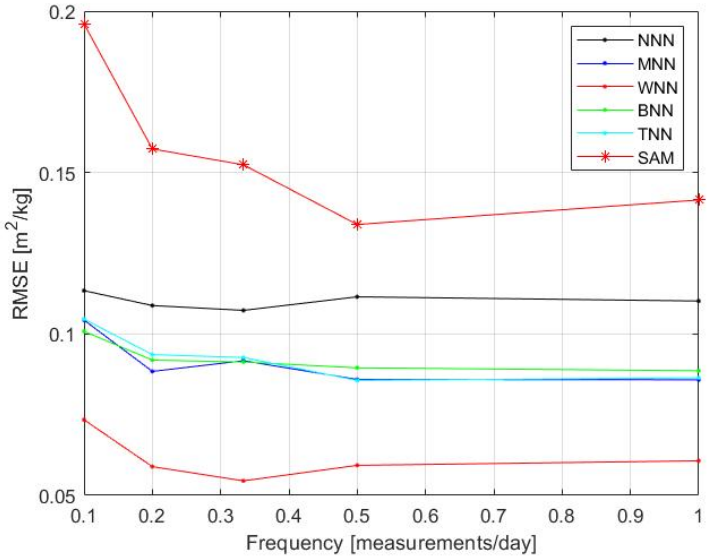


Figure 3.25 - Mean percentage error with respect to the propagation time.

Differently from the NNs, that are quite robust to the number of input measurements, the SAM exhibits a significant increase of the RMSE when the frequency is lower than 2 measurements per day. This might be due to the computation of the time derivatives that becomes more approximative when the frequency decreases.

Performance assessment on real test cases

TLE data from real space objects are used to obtain additional test cases to compare the two approaches. The set of objects is composed of 3 debris with initial orbital elements as reported in Table 3.16. These values are extracted from TLEs taken from SpaceTrack [114] and refer to the first available TLE in January 2021. Then, all the available TLEs within a time interval of 30 days have been considered for the *BC* estimation. For each RSO, the ballistic coefficient is estimated with every neural network and the semi-analytical method (see Table 3.17). The object is then propagated with these predicted values and the semimajor axis compared with the one derived from the TLEs (which is used as a benchmark). The maximum Δa (Δa_{max}) is computed for each neural network and the SAM and reported in Table 3.18. It can be noticed that, in most of the cases, the Δa_{max} is smaller than 1 km after 30 days of propagation. Moreover, the SAM exhibits a Δa_{max} similar to the one of the NNs.

Figure 3.26 shows the semimajor axis obtained using the predicted *BC* with respect to the one obtained from the TLEs.

It must be noted that, even if the WNN is the NN that better performs with the numerical results, it doesn't lead to the smallest Δa_{max}. This might depend on the effect of the uncertainty on the atmospheric density. The error that is due to the WNN might have the same sign of the error due to the atmospheric density and the two effects might sum up.

Table 3.16 - Initial orbital elements of the Resident Space Objects.

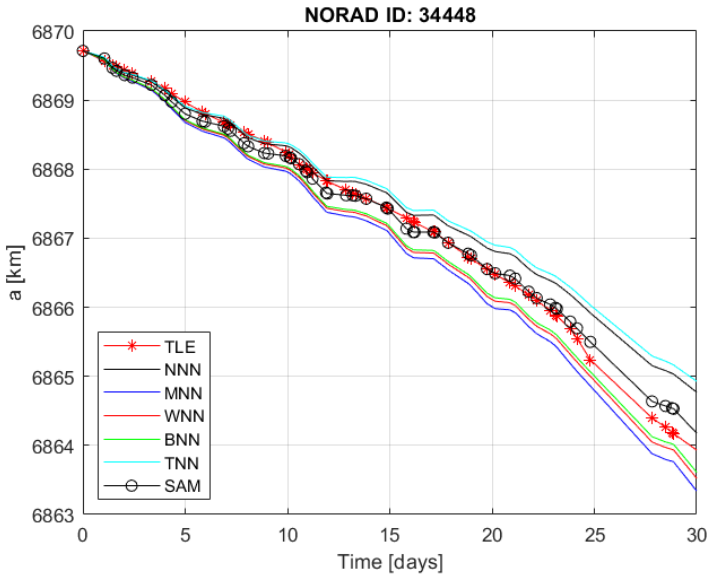
Object NORAD ID	a $[km]$	e [−]	i [°]	Ω [°]	ω [°]	ν [°]
34448	6878.59	1.44e-4	73.94	234.74	80.82	279.25
34631	6853.67	1.52e-4	74.12	92.52	355.73	4.27
40326	6900.82	2.06e-3	86.47	41.38	82.09	277.82

Table 3.17 - Predicted BC for Resident Space Objects.

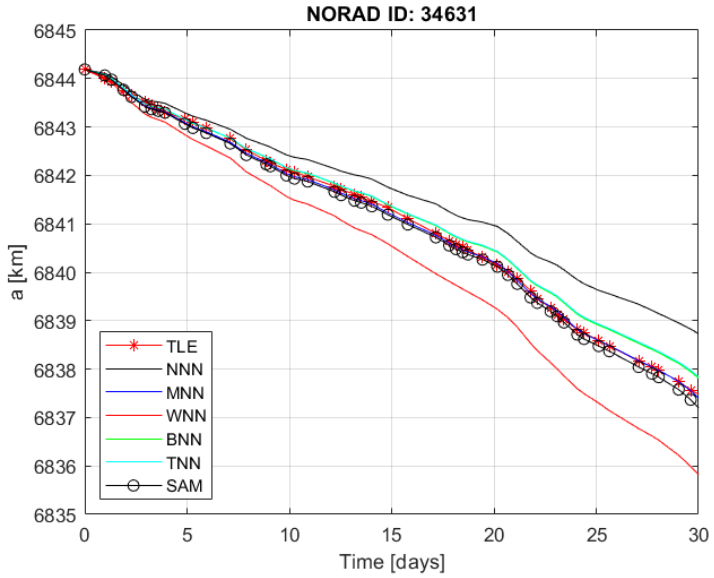
	Predicted BC [m^2/kg]					
Object NORAD ID	NNN	MNN	WNN	BNN	TNN	SAM
34448	0.449	0.565	0.550	0.543	0.436	0.575
34631	0.288	0.351	0.428	0.332	0.330	0.412
40326	1.205	1.138	1.124	1.193	1.301	1.486

Table 3.18 - Δa_{max} for Resident Space Objects.

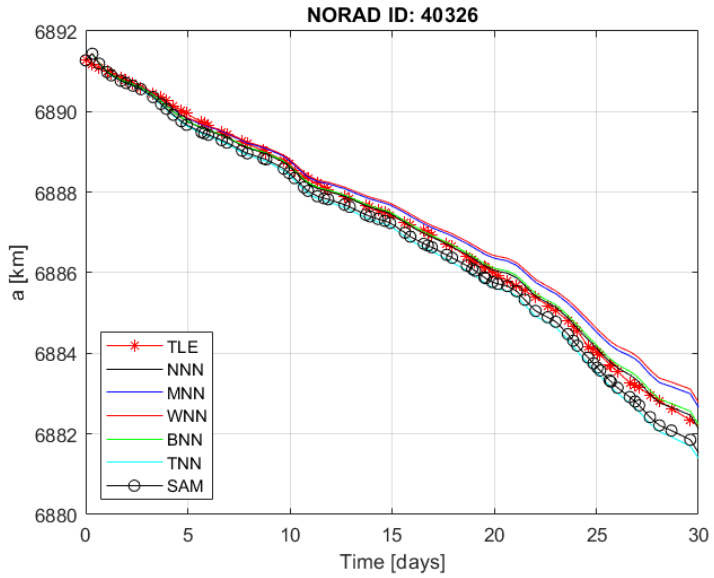
Object NORAD ID	Δa_{max} $[km]$					
	NNN	MNN	WNN	BNN	TNN	SAM
34448	0.861	0.610	0.446	0.410	0.996	0.359
34631	1.263	0.110	1.684	0.383	0.411	0.308
40326	0.262	0.668	0.779	0.282	0.957	0.835



(a)



(b)



(c)

Figure 3.26 - Trend of the semimajor axis versus time for objects with NORAD ID 34448 (a), 34631 (b) and 40326 (c).

3.2.5 Improving the *BC* estimation

Starting from the results previously presented, this section provides two main contributions aiming at improving the performance of the proposed methodology. First, it proposes a reduced set of features to train Neural Networks for ML-based ballistic coefficient estimation of RSOs, while also investigating two preprocessing techniques to reduce the computational cost of the training phase and to filter the astrometric data received in input by the trained network, thus removing undesired oscillations.

Furthermore, the *BC* of a space object can vary significantly over a certain time interval, especially for space objects for which the combination of shape and attitude dynamics determine a significant time variation of their cross section, which has a non-negligible impact on the orbital trajectory. Therefore, the potential of the proposed approach to detect ballistic coefficient variation for space objects in LEO is investigated. A sensitivity analysis is conducted to assess the performance of the proposed approach as a function of the percentage of variation of the *BC* as well as the time at which such a variation takes place.

New datasets generation and preprocessing techniques description

A new training and testing dataset have been generated with six main differences with respect to the methodology presented in Section 3.2.1:

- a different, more computationally efficient, numerical propagator, i.e., THALASSA [130], has been adopted, that integrates and solves the equation of motion in EDromo's formulation [131];
- the values of the semimajor axis and *AMR* have been randomly extracted from uniform distributions within the intervals [400, 600] km and [0.01, 0.60] m^2/kg, and not discretely defined;
- $N_{obj,tr}$ and $N_{obj,te}$ are equal to 3000 and 300, instead of 1650 and 50;

- the propagation has been extended from 30 days to 184 days;
- 7 x 7 harmonics of the EGM2008 gravity model, instead of 20 x 20;
- the atmospheric model is the NRLMSISE-00 instead of Jacchia-Bowman 2008, since the latter was not available in THALASSA;

Space weather is accounted for using data from CelesTrak [75]. Then, as in Section 3.2.1, propagation data are sampled with a frequency of one measurement per day and the consistency error $\boldsymbol{\delta X_{\Delta t}}$ is computed according to Eq. 3.1. Out of the five Neural Networks presented in Section 3.2.1 only two of them have been, namely the Trilayered and the Wide Neural Networks, being the ones better performing. For the validation phase, always a 20% of the training set is used.

In this section, a further step is introduced to process the training and testing datasets before giving them as input to the (trained) learner. In particular, two techniques have been investigated, namely a *normalization* of the data and a *moving mean*.

Normalizing the dataset is a common practice in machine learning to improve algorithm convergence, computational cost and, sometimes, performance [132]. However, it is important to note that not all machine learning algorithms require normalization, and the necessity of normalization should be considered on a case-by-case basis. Hence, the effect of normalizing the data on the proposed algorithm is here analysed. In particular, out of the whole set of features used to train the NN (i.e., initial orbital elements OEl_0, the variation δOEl and the propagation time Δt), the initial orbital elements have been normalized according to the following equation.

$$\bar{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.13}$$

where x is the actual value of the variable, and x_{max} and x_{min} assume the values reported in Table 3.19 for the orbital parameters (semimajor axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω, and mean anomaly M) and the Δt. In Table 3.19, R is the Earth mean equatorial radius,

equal to 6378.1363 km. For the consistency errors instead, the following equation has been used. This definition avoids negative values for the normalized semimajor axis.

$$\bar{x} = \frac{x}{x_{max} - x_{min}} \tag{3.14}$$

Table 3.19 - Values of the orbital parameters and Δt for input data normalization.

	a [km]	e [-]	i [°]	Ω [°]	ω [°]	M [°]	Δt [days]
x_{min}	R	0	0	0	0	0	1
x_{max}	R+650	0.1	360	360	360	360	184

A moving mean allows to reduce undesired oscillations in the time evolution of orbital parameters, improving the phenomenon observability. Defined a vector $\boldsymbol{V}$ of N_v elements, the moving mean applied to $\boldsymbol{V}$ returns an array of N_v local k-point mean values, where each mean is calculated over a sliding window of length k across neighbouring elements of $\boldsymbol{V}$. When k is odd, the window is centred about the element in the current position. When k is even, the window is centred about the current and previous elements. The window size is automatically truncated at the endpoints when there are not enough elements to fill the window. When the window is truncated, the average is taken over only the elements that fill the window. In this work, the moving mean has been applied to the consistency errors of each object of the training/testing dataset. The effect of k over the performance has been analysed.

Numerical results with constant *BC*

The performance of the proposed methodology considering the new datasets has thus been assessed on the 3000 (training dataset) and 300 (testing dataset) synthetic objects, generated as described before. The initial epoch for the orbital propagation is the 1st January 2021 at 00:00:00.000 UTC. The *BC* of the space objects is constant (i.e., equal to its initial value) for the whole observation time.

The objects of the testing dataset are split into three subsets:

- Low-BC Set (LS): $0.022 \leq BC < 0.1$ m^2/kg
- Medium-BC Set (MS): $0.1 \leq BC < 1$ m^2/kg
- High-BC Set (HS): $1 \leq BC < 1.32$ m^2/kg

In fact, in Section 3.2.2 two types of objects were not characterised correctly: objects that re-enter in the atmosphere or that are characterized by a very low *BC* (i.e., smaller than 0.08 m^2/kg). A similar behaviour is expected with the new datasets and, therefore, the performance metrics for the three subsets are evaluated separately.

For each subset, the mean, the median and the standard deviation of the absolute value of the percentage error, computed as in Eq. 3.8, is reported in Table 3.20. In particular, it must be noted that the LS, MS and HS subsets represent the 6.0%, 69.0% and 25.0% of the whole dataset respectively. The WNN exhibits better performance than the TNN for the LS and the MS both in terms of mean and median of the percentage error. On the contrary, the TNN outperforms the WNN for the HS subset (and hence for the objects that most likely re-enter in the atmosphere). These results are in agreement with the ones presented in Section 3.2.2. It is important to highlight that the large percentage error observed in the LS subset is partly attributable to the inherent definition of the error, which involves dividing the absolute error by a relatively small value. Table 3.20 reports in brackets the computational cost of the training phase which refers to a single training.

Table 3.20 - Simulated test case: mean, median and standard deviation of the absolute percentage error for the Low-BC (LS), Medium-BC (MS) and High-BC (HS) subsets. The average computational cost is reported in brackets.

		μ [%]	m [%]	σ [%]
TNN (214 s)	LS	256.1	216.2	192.5
	MS	20.3	8.4	31.5
	HS	2.7	2.2	1.8
WNN (659 s)	LS	224.0	167.0	232.5
	MS	17.3	5.8	31.9
	HS	3.7	2.7	3.2

Performance evaluation with a reduced set of features

In Section 3.2.1, a sensitivity analysis with respect to different sets of features has led to the choice of a particular set, i.e., initial mean orbital elements, the consistency errors δOEl of all the mean orbital elements at a propagation time Δt and the propagation time Δt, as also mentioned before. The performance reported in Table 3.20 have been obtained with such a set.

However, the two orbital parameters mostly affected by the atmospheric drag, and hence by the *BC*, are the semimajor axis and the eccentricity. Therefore, in this section a reduced set of features is presented and the performance in terms of accuracy and computational time is compared. In fact, reducing the features number normally leads to a reduced computational cost [132]. The new set of features consists of the $OEl_0, \delta a,\ \delta e$ and Δt. Table 3.21 reports the performance metrics and the computational cost. It can be noticed that for the TNN the error metrics improve of few percentage points with equal computational cost. For the WNN instead, there is a slight reduction of the LS mean and a decrease of 17% of the computational cost, while other metrics get worse. The different outcome of the two NNs is due to the different networks' architectures.

Table 3.21 - Simulated test case with a new set of features: mean, median and standard deviation of the absolute percentage error in simulated test cases, for the Low-BC (LS), Medium-BC (MS) and High-BC (HS) subsets. The average computational cost is reported in brackets.

		μ [%]	m [%]	σ [%]
TNN (217 s)	LS	245.3	207.3	189.5
	MS	20.3	7.8	32.0
	HS	2.4	1.6	2.0
WNN (547 s)	LS	214.0	182.7	207.9
	MS	20.4	8.9	34.3
	HS	3.9	2.6	3.6

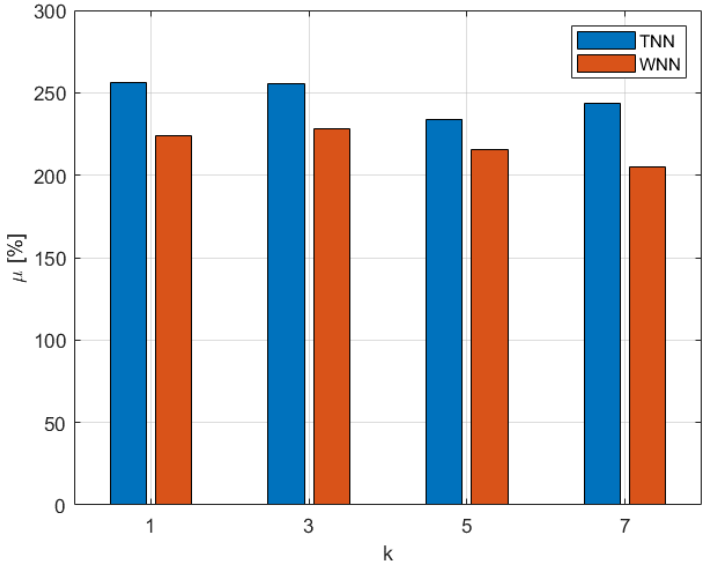
Input data preprocessing

Two different data preprocessing techniques have been analysed separately. First, the input data have been normalized according to Eq. (3.14) and (3.15). The performance metrics are reported in Table 3.22. It can be noted that the performance is different for the three subsets. In fact, for the HS, both the mean and the median of the percentage error decrease. For the LS and the MS, instead, the mean and median follow different trends. For both the TNN and WNN, the computational cost decreases, according to [132], of 12.6% and 18.8% respectively.

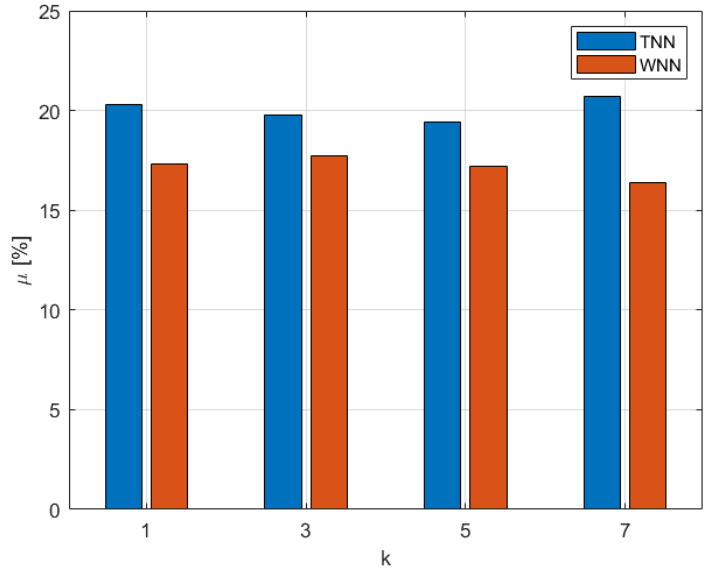
Table 3.22 - Simulated test case when normalizing the input data: mean, median and standard deviation of the absolute percentage error in simulated test cases, for the Low-BC (LS), Medium-BC (MS) and High-BC (HS) subsets. The computational cost is reported in brackets.

		μ [%]	m [%]	σ [%]
	LS	255.7	231.4	197.4
TNN (187 s)	MS	19.9	7.9	32.0
	HS	2.2	1.6	1.8
	LS	222.9	153.6	217.1
WNN (535 s)	MS	18.3	6.0	33.0
	HS	3.4	2.4	3.1

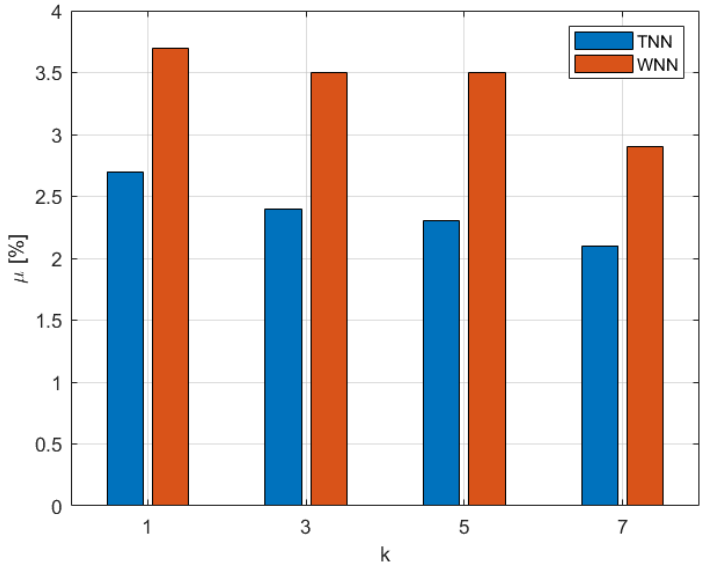
The second technique being assessed is the moving mean, as described before. Three different values of k (i.e., number of neighbouring points over which the mean is calculated) have been defined, i.e., 3, 5 and 7 points, corresponding to a time window of 3, 5 and 7 days respectively. Figure 3.27 shows the mean percentage error as a function of k, for the LS (a), MS (b) and HS (c). In all the cases, there is at least one value of k that leads to an improvement of the performance with respect to the results obtained without applying the moving mean (i.e., k equal to 1). In particular, for the TNN this value is 5 for the LS and MS, while it is 7 for the HS. For the WNN, instead, a moving mean over a time window of 7 days leads to a reduction of the mean percentage error for all the objects of the testing dataset.


300
250
200
150
100
50
0
μ [%]
1
3
5
7
k
TNN
WNN

(a)


25
20
15
10
5
0
μ [%]
1
3
5
7
k
TNN
WNN

(b)



(c)

Figure 3.27 - Bar plot of the mean percentage error using the moving mean on the input data for the LS (a), MS (b) and HS (c) subsets. *k* equal to 1 represents the case where no moving mean is applied.

Numerical results with time-varying BC

In the previous analyses the *BC* was assumed to be constant over the whole time of interest. Even if this can be a good assumption for a wide set of Earth orbiting objects, as explained in Section 3.2.1, there are several scenarios in which a sudden change of the *BC* can occur. For instance: an operative satellite can deploy one or more solar panels increasing significantly the frontal section area; a satellite can change its attitude (and hence the cross-section area) due to a different operational scenario; as a consequence of a breakup event, an object could change its shape, mass and surface.

Therefore, in this section, a step function for the *BC* has been assumed, according to the following law:

$$BC = \begin{cases} BC_0 & if\ t < t_1 \\ BC_0 + \dfrac{pc}{100} * BC_0 & if\ t \geq t_1 \end{cases} \tag{3.15}$$

where BC_0 represents the object's initial Ballistic Coefficient, pc denotes the percentage by which the *BC* changes, and t_1 is the time instant when this change occurs. It is important to highlight that a variation in the value of *BC* should be large enough to generate observable impact on the evolution of orbital parameters. In this respect, a sensitivity analysis in terms of t_1 and pc is carried out. In particular, t_1 and pc are integer numbers and can assume the following values: {30, 90, 150} days and {+50, +100} respectively. For pc, only positive values have been taken into account to increase the *BC* change observability.

For each possible couple (t_1, pc), all the objects of the testing data set have been propagated for 184 days (or until re-entry). The difference between the mean semimajor axis obtained with the BC variation ($a_{\Delta BC}$) and the one obtained with no *BC* variation (a_{ref}) is computed for each object of the testing dataset:

$$\delta a_{\Delta \mathrm{BC}} = a_{\Delta \mathrm{BC}} - a_{ref} \tag{3.16}$$

Table 3.23 reports the 25th percentile (Q_1), the 75th percentile (Q_3), the mean and the median of the $\delta a_{\Delta BC}$ distribution evaluated at the 184th day. As expected, the $\delta a_{\Delta BC}$ assumes the maximum value of the mean percentage error, i.e. -43.6 km, for the pair (30, 100). Figure 3.28 and Figure 3.29 show the same metrics over time for t_1 equal to 30 days and for pc equal to +50% and +100% respectively. Clearly, the $\delta a_{\Delta BC}$ always assumes negative values for $t > t_1$ since a greater value of the *BC* entails a greater decay of the object. The overall trend of $\delta a_{\Delta BC}$ is decreasing. The strongly oscillating behaviour of the mean value is due to the presence of some objects that re-enter in the observation time and thus introduce very large values of $\delta a_{\Delta BC}$.

Table 3.23 - 25th (Q_1) and 75th (Q_3) percentiles, mean and median of $\delta a_{\Delta BC}$ distribution evaluated at the 184th day, for (t_1, pc) equal to (30, 100).

pc	t_1 [days]	Q_1 [km]	Q_3 [km]	μ [km]	m [km]
50	30	-25.2	-5.8	-21.3	-12.5
	90	-16.0	-3.6	-14.8	-7.8
	150	-5.0	-1.1	-6.0	-2.5
100	30	-44.2	-10.6	-43.6	-21.9
	90	-29.4	-6.8	-26.6	-14.7
	150	-10.0	-2.1	-15.8	-4.8

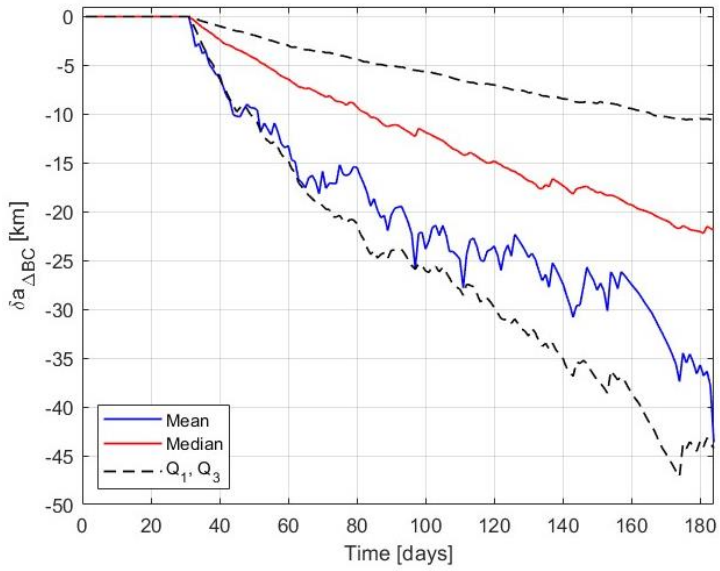


Figure 3.28 - 25th (Q_1) and 75th (Q_3) percentiles, mean and median of $\delta a_{\Delta BC}$ distribution as a function of time, for (t_1, pc) equal to (30, 100).

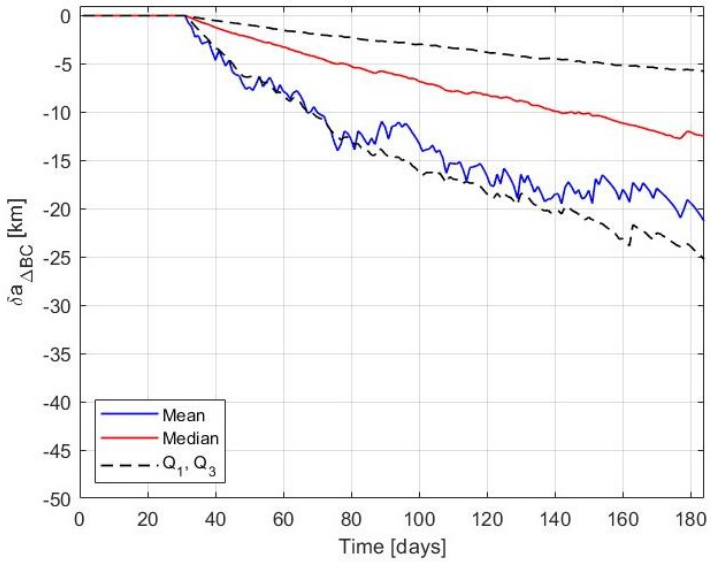


Figure 3.29 - 25th (Q_1) and 75th (Q_3) percentiles, mean and median of $\delta a_{\Delta BC}$ distribution as a function of time, for (t_1, pc) equal to (30, 50).

Such analysis provides an overview of the expected δa values for the testing dataset objects. Evaluating the performance of a sample object, that exhibits a certain behaviour in terms of δa as a function of pc and t_1, allows predicting the performance of likewise objects.

Therefore, to demonstrate how the neural networks here presented behave in case of a ΔBC, the BC has been estimated for each object of the testing dataset for which the condition:

$$BC_0 + \frac{pc}{100} BC_0 < 1.32 \frac{m^2}{kg} \tag{3.17}$$

is satisfied, being 1.32 m^2/kg the maximum value of the BC used for the training dataset. In particular, the results for a specific object of the dataset are reported and shown in Figure 3.30 and Figure 3.31. This object has the initial mean orbital elements reported in Table 3.24 and a BC_0 equal to 0.6503 m^2/kg.

Table 3.24 – Initial mean orbital parameters of a sample object of the testing dataset

a [km]	e [-]	i [°]	Ω [°]	ω [°]	M [°]
6912.35	$5.63\ 10^{-3}$	76.6	26.6	177.4	76.3

Figure 3.30 shows the time evolution of the mean semimajor axis for *BC* equal to BC_0 (i.e., a_{ref}) and *BC* increased of 100% for t_1 equal to 30, 90 and 150 days. This leads to a δa at the last propagation day of 232.5 km, 51.6 km and 13.8 km respectively. For each case, the estimated *BC* is shown in Figure 3.31 both for the TNN and the WNN, and compared with the reference value (BC_{ref}). It can be noticed that, in all the cases, the estimated *BC* starts to increase for $t > t_1$ and gets closer to the reference value. Clearly, for t_1 = 30 days the performance is better as the observability window is larger than the other two cases. In particular, the error of the predicted *BC* is lower than 3% on the last propagation day. For pc equal to 50%, the results obtained are shown in Figure 3.32 and Figure 3.33. The δa at the end of the propagation is equal to 39.5 km, 20.2 km and 6.6 km for t_1 equal to 30, 90 and 150 days respectively (see Figure 3.32). Also in this case, the estimated *BC* starts to increase for $t > t_1$ and gets closer to the reference value (Figure 3.33). In particular, for t_1 = 30 days the error of the predicted *BC* is lower than 8 % on the last propagation day. Similar results are obtained for objects with a $0.2 < BC_0 < 0.65$ m^2/kg. On the contrary, for $BC_0 < 0.2$ m^2/kg, two trends have been observed: either the *BC* starts to increase at a time much larger than t_1 (e.g., after 90 days) or the neural network does not detect any variation of the *BC* in the observation window.

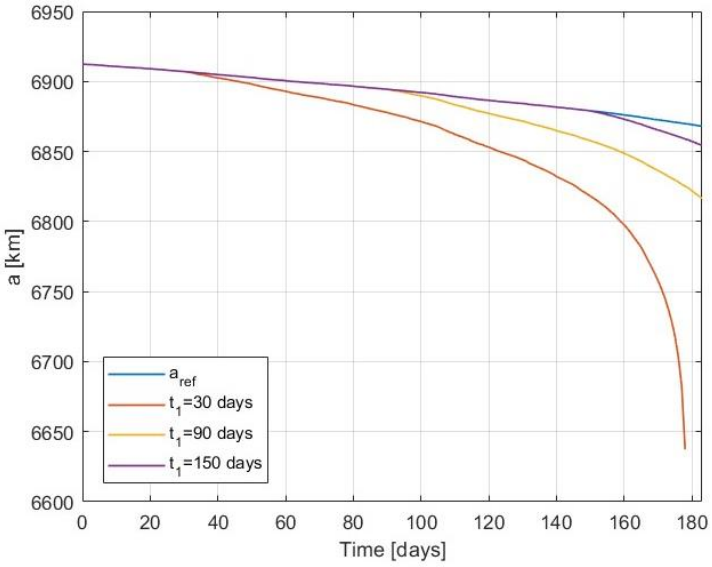
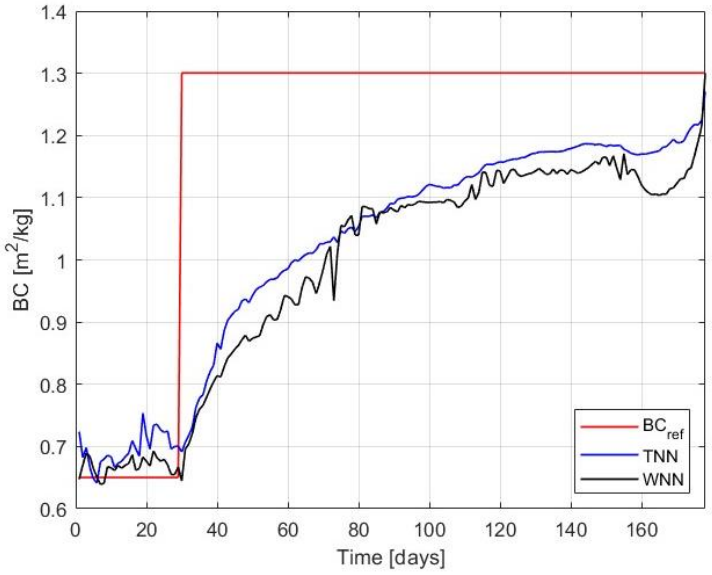
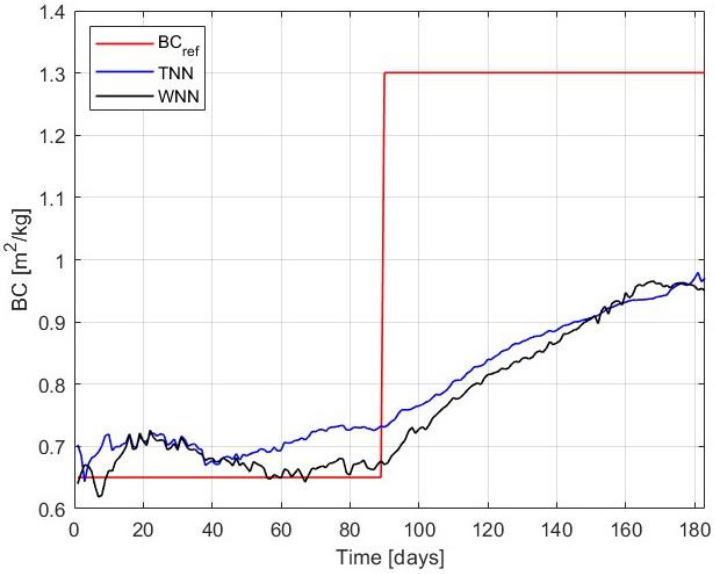


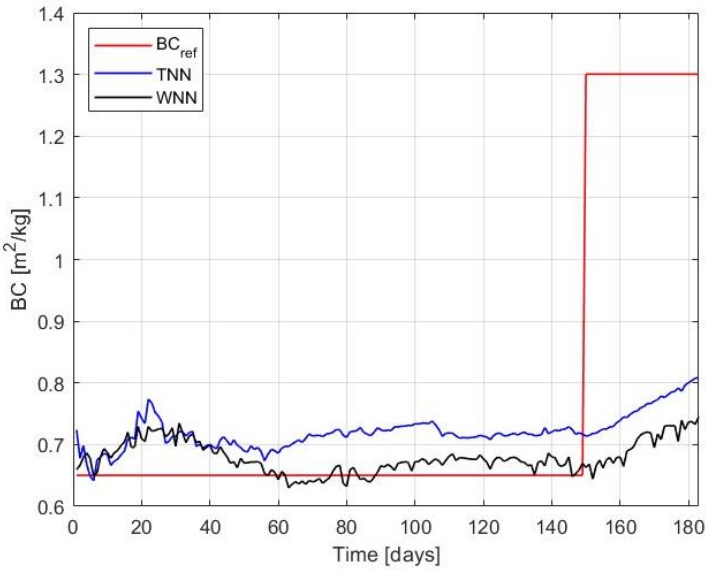
Figure 3.30 - Time evolution of the semimajor axis with constant *BC* (a_{ref}) and with a *BC* variation of 100% for t_1 equal to 30, 90, and 150 days.



(a)



(b)



(c)

Figure 3.31 - Reference and predicted *BC* of the sample object for a *BC* variation of 100% at t_1 equal to 30 (a), 90 (b) and 150 (c) days.

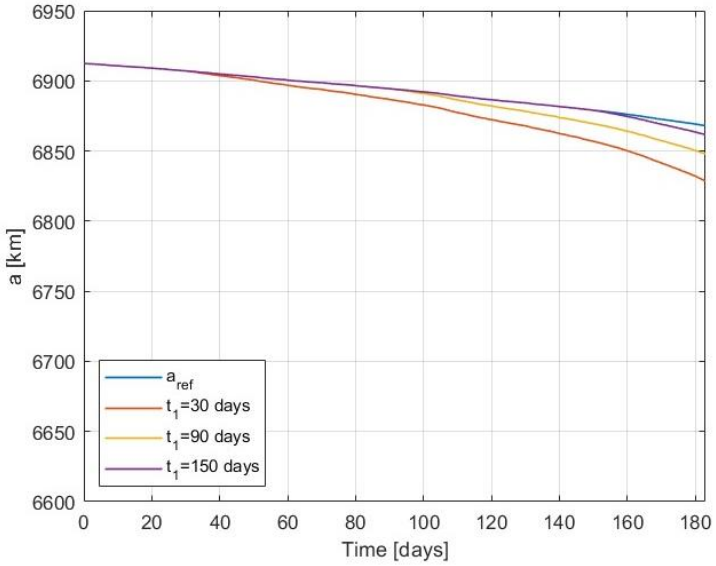
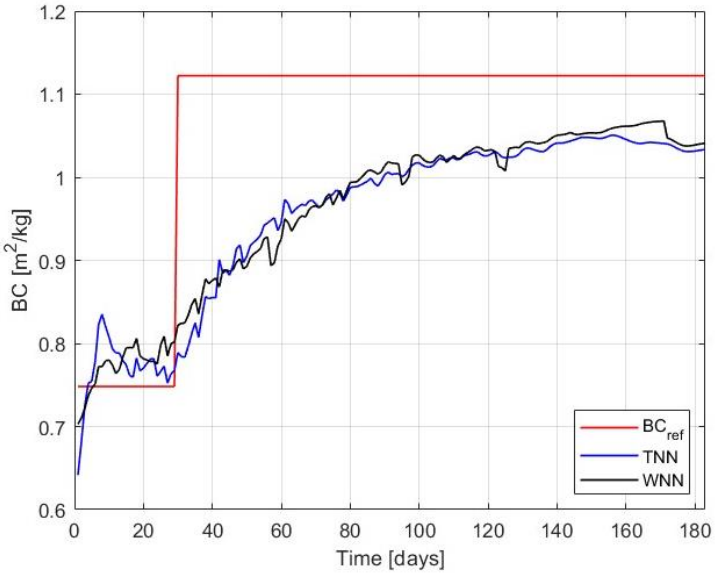
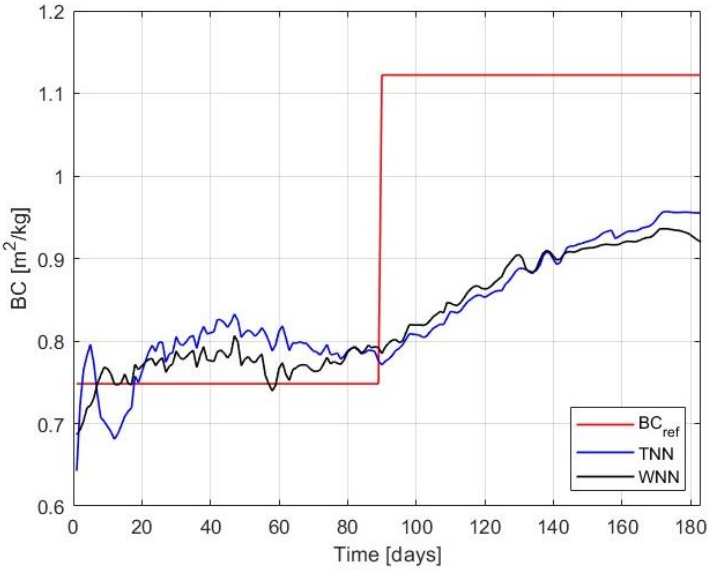


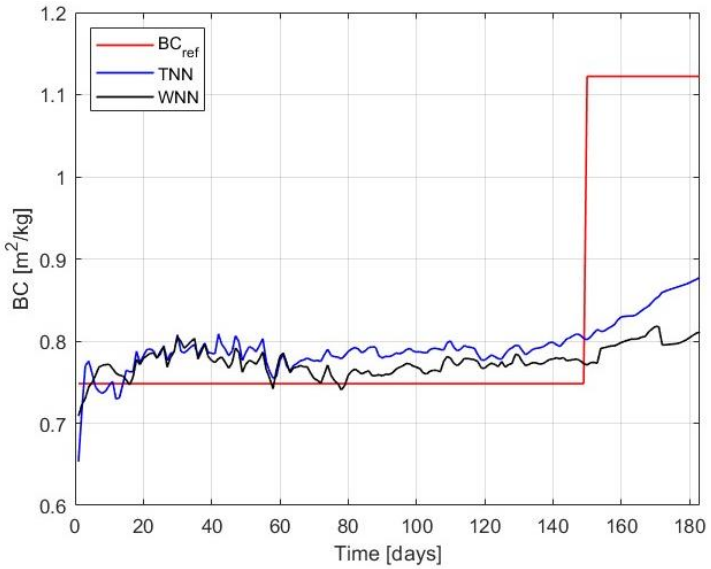
Figure 3.32 - Time evolution of the semimajor axis with constant *BC* (a_{ref}) and with a *BC* variation of 50% for t_1 equal to 30, 90, and 150 days.



(a)



(b)



(c)

Figure 3.33 - Reference and predicted *BC* of the sample object for a *BC* variation of 50% at t_1 equal to 30 (a), 90 (b) and 150 (c) days.

3.3 Recurrent Neural Networks

The methodology proposed in Section 3.2.1 has three shortcomings. First, the estimation of the *BC* at a certain time instant is independent from previous information (i.e., the time evolution of the orbital parameters before that time): this aspect is due to the intrinsic structure of feedforward neural networks. Second, the algorithm can be very sensitive to the uncertainty in the initial conditions of the RSO, especially for lower altitudes, as it requires the propagation of the initial state (and hence of the related uncertainty) for an extended period of time. Finally, the generation of the training set entails propagating the same synthetic object twice: once with the actual *BC* and once with a default value: this need can lead to a high computational cost for the generation of the training dataset.

With this regard, this section investigates the use of Recurrent Neural Networks (RNNs) for the estimation of the *BC* of RSOs in LEO, using astrometric data. In fact, unlike feedforward neural networks, RNNs are optimised for time-series data, leveraging the underlying information in the time evolution of the orbital parameters. A numerical environment is used for the training, validation and testing phase, including realistic orbit determination errors in the generation of the simulated astrometric data. A hyperparameters tuning and a features selection process are performed to design the RNN architecture. Moreover, a sensitivity analysis is carried out to assess the variability in the performance of the proposed methodology against change in the number of objects in the training dataset, the propagation time, and the uncertainty in the atmospheric density model. Finally, additional analyses are carried out to evaluate the robustness against measurement noise in the input data, and considering a time-varying data sampling step. This latter point is particularly important since RNNs typically work on evenly time-spaced data, which is an assumption that is usually not verified for real astrometric data (e.g., Two-Line Elements).

3.3.1 Recurrent Neural Network architecture

Recurrent Neural Networks are a type of artificial neural network designed to effectively process sequential data [132]. Unlike classical feedforward neural networks, RNNs have characteristics which allow them to maintain a memory of past inputs, making them well-suited for tasks involving sequences, time-series data, and natural language processing. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are two advanced architectures of Recurrent Neural Networks (RNNs) designed to address the vanishing gradient problem and improve the modelling of long-term dependencies in sequential data. The main difference between these two architectures lies in their internal structure. LSTMs have a more complex design which allows capturing and controlling information flow more precisely over extended sequences. On the other hand, GRUs have a simplified structure being computationally less expensive. In this work, GRU has been used and its detailed structure is described in the following subsection.

Regardless of the neural network architecture, the first phase of any supervised machine learning method consists in generating the training, validation and testing datasets. In this work, the training set consists of orbital data from a number of synthetic space objects with known *BC*. To build such training set, the following steps have been followed.

First, the intervals of orbital parameters and *AMR* values to consider for the analyses have been defined. These intervals, are the same as in Table 3.1. This choice also allows to compare performance of RNNs for space object characterization against previous results based on feedforward neural networks. A set of $N_{obj,tr}$ objects is generated by randomly extracting their initial orbital parameters set (indicated in a vector form as $\boldsymbol{OEl_0}$) and their *AMR* values from uniform distributions inside the aforementioned intervals. Assuming a default C_d equal to 2.2, a reference ballistic coefficient (BC_{true}) is assigned to each object.

Each of the synthetic objects is propagated for a time period of N_{days} days, which is set equal to 180 days. THALASSA [130] is adopted as numerical propagator, that integrates and solves the equation of motion in EDromo's formulation [131]. The

following perturbations are considered: 7 x 7 harmonics of the EGM2008 gravity model [111], aerodynamic drag using the exponential atmospheric density (ρ) model [133] reported in Eq. 3.18, third-body perturbations due to the Sun and the Moon, and a cannonball model for Solar Radiation Pressure (SRP) [134]. In Eq. 3.18, ρ_0 is the density at a reference altitude h_0 and H the scale height. The reference density ρ_0 and scale height H are piecewise continuous, as reported in [113].

$$\rho = \rho_0 \exp\left(-\frac{h - h_0}{H}\right) \tag{3.18}$$

The *BC* is assumed to be constant over the whole time of interest. The propagated orbital states, i.e., position and velocity components expressed in the Earth Centred Inertial (ECI) reference frame, are then sampled with a frequency of one measurement per day. The resulting position and velocity vectors are converted into osculating orbital parameters and then into mean orbital parameters by removing short-periodic terms according to Brouwer's theory [113]. Since in Section 3.2.1 mean orbital parameters have been demonstrated to provide better performance than osculating ones in the training and testing phases, also in this paper they have been chosen for the subsequent steps of the proposed methodology.

The time series of mean orbital parameters obtained through this procedure is further pre-processed by applying *normalization* and *padding* techniques, whose purpose, methodology and implementation details are provided in the following subsection. A percentage of the training set, equal to 20% as in [115], is used for the validation phase. The flow diagram of the training and validation phase is shown in Figure 3.34, in which $\boldsymbol{X}$ is a matrix containing the time-series data of sampled mean orbital elements, whereas $\overline{\boldsymbol{X}}$ is the matrix obtained normalizing and padding $\boldsymbol{X}$. Both $\boldsymbol{X}$ and $\overline{\boldsymbol{X}}$ are described more in detail in the following subsection.

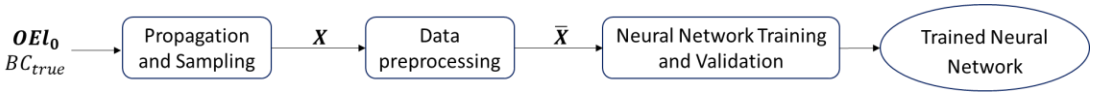


Figure 3.34 - Flow diagram of the training and validation phase.

A crucial step for the learning process is the choice of the features as they must capture the relation between the outcome of the ML algorithm and the input data. With this regard, in Section 3.2.1 a sensitivity analysis with respect to different sets of features has led to the choice of a particular set, i.e., initial mean orbital elements, the consistency errors $\delta\boldsymbol{OEl}$ of all the mean orbital elements at a propagation time Δt and the propagation time Δt. Similarly in this section, six mean orbital elements have been chosen as features: semimajor axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω, and mean anomaly M. It is important to highlight that the propagation time is not necessary as a feature when using RNNs, since it is an intrinsic information in equally spaced time-series data.

Once the learner is trained and validated, the proposed approach performance can be assessed using a testing dataset, which is built as an ensemble of $N_{obj,te}$ objects whose orbital parameters and *AMR* values are randomly extracted from uniform distributions defined in the same intervals as those used to generate the training dataset (Table 3.1). Assuming again C_d = 2.2, a true value of *BC* (BC_{true}) can be assigned to each element of the testing dataset. Each element undergoes the same steps of propagation, sampling and data preprocessing used to generate the training dataset. Measurements noise will also be considered in Section 3.3.2. The data thus obtained, $\boldsymbol{OEl}$ feeds the trained neural network and a *BC* is predicted (BC_{pred}). The root mean square error (*RMSE*) between BC_{pred} and BC_{true} is used as performance metric, and is computed as in Eq. 3.2. The mean (μ), median (m) and standard deviation (σ) of the percentage errors distribution over the elements of the dataset, computed as in Eq. 3.8, are used as further performance metrics. The flow diagram of the testing phase is shown in Figure 3.35.

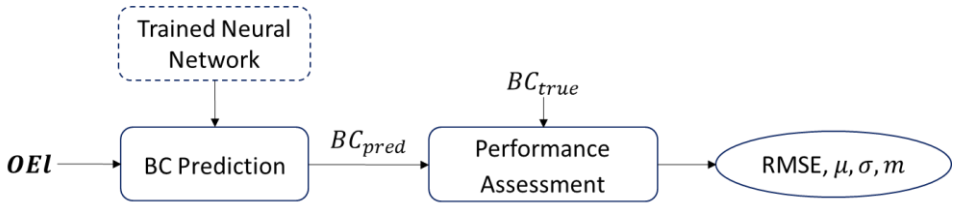


Figure 3.35 - Flow diagram of the testing phase.

Neural Network Design

The trajectory of a space object can be fully described by the time evolution of a set of six scalars, either in the form of Cartesian coordinates (i.e., position and velocity) or as a set of orbital elements. After having sampled the trajectory, a time series of n measurements is available. Therefore, being f the number of features used for the ML technique (i.e., f equal to 6), an individual feature sequence (i.e., relative to one object of the training dataset) feeding the neural network (NN) during the training can be represented as an $[n\ x\ f]$ matrix:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1f} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nf} \end{pmatrix} \tag{3.19}$$

where $\boldsymbol{x}_i$ is the vector containing the mean orbital parameters at the i-th sampling time, and $\boldsymbol{X}$ represents the matrix containing the sequence of sampled mean orbital elements over N_{days}**.** This matrix undergoes two preprocessing techniques. First, to enhance the training convergence, the network features are normalized according to

$$\bar{x}_{ij} = \frac{x_{ij} - x_{min,j}}{x_{max,j} - x_{min,j}} \tag{3.20}$$

where x_{ij} is the actual value of the variable, and $x_{max,j}$ and $x_{min,j}$ are fixed predefined values for the j-th feature.

Moreover, some objects of the training (and testing) dataset might re-enter in the atmosphere within N_{days} and the value of n will thus be different across objects. Therefore, to ensure consistent size of the input data for the neural network, the normalized matrix of these objects is *padded*, i.e., extra rows of 0 values are added at the end of the matrix. The resulting matrix, depicted as $\bar{\boldsymbol{X}}$ in Figure 3.34, and the following variables will be denoted, hereinafter, without the upper bar to simplify notation.

The designed neural network, shown in Figure 3.36, is described as follows:

1. The first layer consists of a feedforward fully-connected layer (FF1) of q neurons. The output of this layer is an $[n\ x\ q]$ matrix:

$$\boldsymbol{X}_{FF1,0} = \boldsymbol{X}\boldsymbol{W}_{FF1} + \boldsymbol{B}_{FF1} \tag{3.21}$$

where $\boldsymbol{W}_{FF1}$ is the weight matrix of size $[f\ x\ q]$ where each column corresponds to the weights for a single neuron; $\boldsymbol{B}_{FF1}$ is the bias vector of size $[1 \times q]$ where each element is the bias term for a single neuron. Before being added, $\boldsymbol{B}_{FF1}$ is broadcasted across the rows of the matrix $\boldsymbol{X}\boldsymbol{W}_{FF1}$.

2. The output from FF1 is combined with the original features at each time instant, thus obtaining an $[n\ x\ (q + f)]$ matrix:

$$\boldsymbol{X}_{FF1} = [\boldsymbol{X}_{FFL1,0}, \boldsymbol{X}] \tag{3.22}$$

3. The new matrix is passed to a Gated Recurrent Unit layer (GRU) of n cells and h units, in a sequence-to-one configuration [132]. Each of the n cells receives as input a feature vector referred to a time instant t, $\boldsymbol{x}_t^{FF1}$ (i.e., a row of $\boldsymbol{X}_{FF1}$), and a previous hidden state vector $\boldsymbol{h}_{t-1}$ from the preceding cell, corresponding to the time instant $t - 1$. This vector contains information flowing from the previous cell to the current. The output of each of the n cells is a hidden state vector $\boldsymbol{h}_t$, transferred to the following cell. To compute $\boldsymbol{h}_t$, a cell relies on two gates: the *reset* gate and the *update* gate. The reset gate $\boldsymbol{r}_t$ influences how much of the past hidden state $\boldsymbol{h}_{t-1}$ should be "reset" or forgotten when computing the candidate hidden state $\overline{\boldsymbol{h}}_t$ for the current time step t. The reset gate enables the GRU to selectively remember or forget information from the past, contributing to the model's ability to capture long-term dependencies in sequential data. The reset gate is a $[h\ x\ 1]$ vector computed as:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_{xr}\boldsymbol{x}'_t + \boldsymbol{W}_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{b}_r) \tag{3.23}$$

where $\boldsymbol{W}_{xr}$ and $\boldsymbol{W}_{hr}$ are weight matrices of dimension $[h\ x\ (q + f)]$ and $[h\ x\ h]$ respectively, $\boldsymbol{x}'_t$ is the $[(q + f)\ x\ 1]$ transpose vector of $\boldsymbol{x}_t$, $\boldsymbol{b}_r$ is the $[h\ x\ 1]$ bias vector, and $\sigma(\bullet)$ is the sigmoid activation function applied elementwise and defined as:

$$\sigma(x) = 1/(1 + e^{-x}) \tag{3.24}$$

For the first cell, since there is not a previously computed hidden state $\boldsymbol{h}_{t-1}$, this is initialized with zeros. Then, the candidate hidden state of $[h\ x\ 1]$ dimension can be computed as:

$$\widetilde{\boldsymbol{h}_t} = \phi(\boldsymbol{W}_{xh}\boldsymbol{x}'_t + \boldsymbol{W}_{hh}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h) \tag{3.25}$$

where $\boldsymbol{W}_{xh}$ and $\boldsymbol{W}_{hh}$ are weight matrices of dimension $[h\ x\ (q+f)]$ and $[h\ x\ h]$ respectively, $\boldsymbol{b}_h$ is the $[h\ x\ 1]$ bias vector, the operator $\odot$ represents the element-wise (or Hadamard) product, and $\phi(\bullet)$ is the hyperbolic activation function applied elementwise and defined as:

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.26}$$

The update gate selects what to transfer from the previous hidden state and what from the current candidate hidden state by computing:

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_{xz}\boldsymbol{x}'_t + \boldsymbol{W}_{hz}\boldsymbol{h}_{t-1} + \boldsymbol{b}_z) \tag{3.27}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \widetilde{\boldsymbol{h}_t} \tag{3.28}$$

where $\boldsymbol{W}_{xz}$ and $\boldsymbol{W}_{hz}$ are weight matrices of dimension $[h\ x\ (q+f)]$ and $[h\ x\ h]$ respectively, $\boldsymbol{b}_z$ is the $[h\ x\ 1]$ bias vector. The hidden state computed by each cell is a $[h\ x\ 1]$ vector. In a sequence-to-one model, only the last computed hidden state $\boldsymbol{h}_n$ is given as output of the GRU.

4. The output of the GRU $\boldsymbol{h}_n$ is given as input to a second feedforward fully-connected layer (FF2) consisting of q neurons. The output of FF2 is a $[q\ x\ 1]$ vector computed as:

$$\boldsymbol{X}_{FF2} = \boldsymbol{W}_{FF2}\boldsymbol{h}_n + \boldsymbol{b}_{FF2} \tag{3.29}$$

where $\boldsymbol{W}_{FF2}$ is the weight matrix of dimension $[q\ x\ h]$ and $\boldsymbol{b}_{FF2}$ is the $[q\ x\ 1]$ bias vector.

5. To improve the generalization of the model, thus preventing overfitting, a dropout layer with a dropout probability p is used. During the dropout process, each element of $\boldsymbol{X}_{FF2}$ is "dropped out" with a certain probability p. If the

element is dropped out, it is replaced by a 0, otherwise the element is kept. The $[q\ x\ 1]$ vector thus obtained is $\boldsymbol{X}_p$.

6. Since the output of the neural network is a scalar (i.e., the estimated BC), the final layer consists of a single fully connected neuron which outputs a scalar value BC_{est}, computed as:

$$BC_{est} = \boldsymbol{W}_{out}\boldsymbol{X}_p + b_{out} \tag{3.30}$$

where $\boldsymbol{W}_{out}$ is the weight matrix of dimension $[1\ x\ q]$ and b_{out} is the scalar bias.

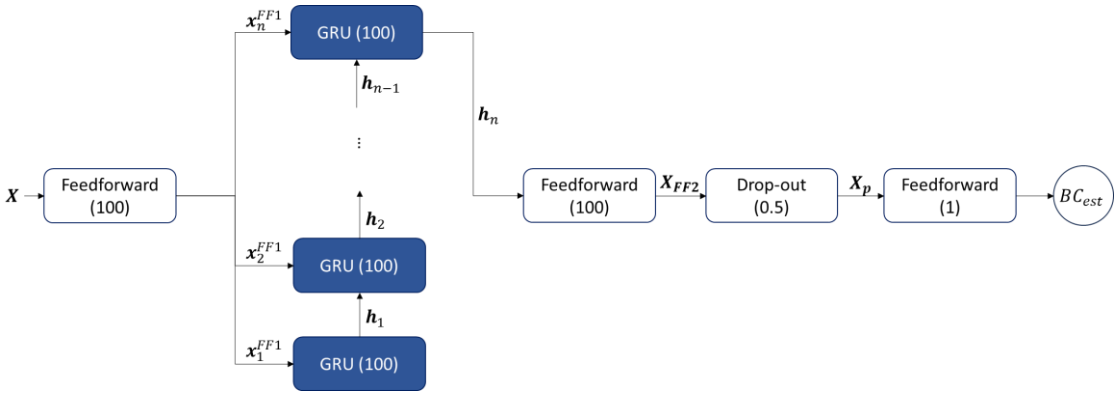


Figure 3.36 - Designed Neural Network. The blue filled blocks represent the GRUs.

Hyperparameter tuning

Hyperparameters are the configuration settings of a model that are not learnt from the data but must be set prior to training. They play a significant role in determining the model's complexity, capacity, and generalization ability. Tuning of the hyperparameters is a crucial step in machine learning and strongly affects the performance of a trained model. In this work, the following hyperparameters have been tuned [36]:

- Number of hidden units of the feedforward layers and of the GRU layer (i.e., q and h respectively).

- Maximum number of training epochs, N_{ep}. A training epoch is a single pass through the entire training dataset. The number of training epochs is a hyperparameter that specifies how many times the algorithm will iterate over the entire training dataset.

- Batch size, S_B. The batch size is the number of training examples used in one iteration of the training algorithm. Instead of updating the model's parameters (i.e., weights and biases) after processing the entire training dataset (i.e., batch gradient descent) or just one example (i.e., stochastic gradient descent), the batch size defines the number of examples used in each update. The number of iterations per epoch is, hence, the total number of training examples divided by the batch size.

- Validation frequency, F_v, i.e., the number of iterations after which the validation is performed.

- Initial learning rate, L_R, i.e., the size of the steps taken during the optimization process of a machine learning model.

- Dropout probability (i.e., p) of the dropout layer.

For each hyperparameter, different values have been considered for the tuning as shown in Table 3.25. The selected value of each hyperparameter, reported in the last column of the table, has been chosen as follows. Starting from an initial set of $q, h, N_{ep}, S_B, F_v, L_R, p$ equal to 100, 50, 30, 25, 500, 0.01, 0.5 respectively, the selected value of the *i*-th hyperparameter is the one minimizing the validation RMSE, keeping fixed the remaining hyperparameters.

Table 3.25 - Set of the hyperparameters chosen for the neural network tuning and selected values.

Hyperparameter	Tuning values	Selected value
q	50, 100, 150, 200, 400	100
h	50, 100, 150	100
N_{ep}	30, 100, 200, 1000, 10000	100

S_B	1, 15, 25, 50, 100	1
F_v	500, 1000	500
L_R	0.001, 0.005, 0.01, 0.02	0.001
p	0.2, 0.5, 0.8	0.5

Among the chosen hyperparameters, a significant role is played by the maximum number of training epochs, the batch size and the initial learning rate. In particular, keeping fixed the other hyperparameters, the RMSE of the training and validation phase decreases of about 50% for a number of epoch greater than 10, a batch size lower than 25, and an initial learning rate smaller than 0.02.

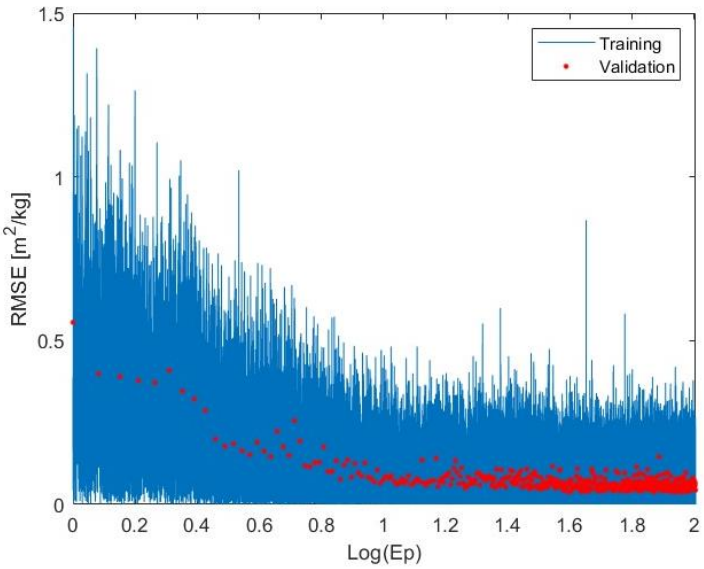
3.3.2 Numerical results

The performance of the proposed methodology has been initially assessed on a set of 3000 (training dataset) and 300 (testing dataset) synthetic objects, generated as described in Section 3.3.1. The initial epoch for the orbital propagation is the 1st January 2021 at 00:00:00.000 UTC. The propagation time is N_{days} = 180 days. Data are normalized according to Eq. 3.13, with the values of $x_{min,j}$ and $x_{max,j}$ reported in Table 3.19.

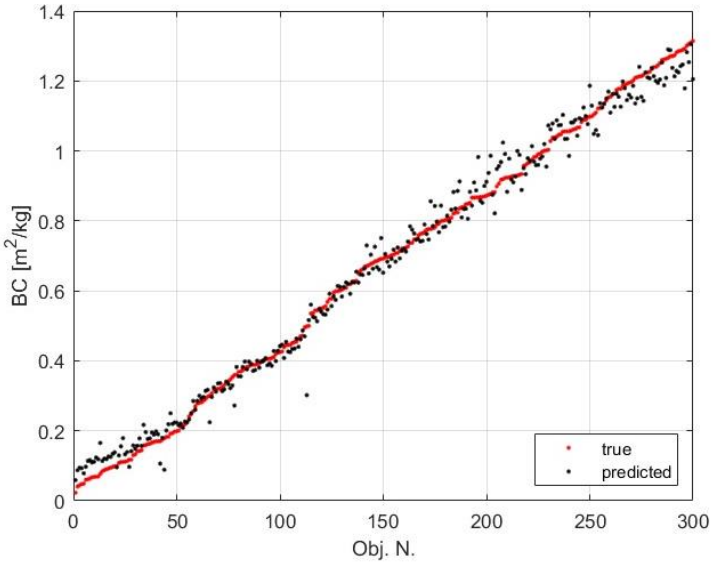
Table 3.26 reports the performance metrics for this test case (TC1), in particular the RMSE on the testing dataset, the RMSE_v of the validation phase, the mean (μ), median (m) and standard deviation (σ) of the absolute percentage error evaluated on the testing dataset, and the computational cost of the training phase. The computational cost of the testing phase is negligible, with the time necessary to estimate the *BC* for the whole testing dataset being less than one second. Figure 3.37 (a) shows the RMSE of the training (blue solid line) and validation (red dots) phases as a function of the training epoch E_p. It can be noticed that both values decrease of about 50% for E_p equal to 100, reaching steady state by the end of the training. Figure 3.37 (b) shows the predicted and the true *BCs* for each object of the testing dataset.

Table 3.26 - Performance metrics for $N_{obj,tr}$ equal to 3000 objects.

$N_{obj,tr}$	RMSE [m²/kg]	RMSE$_v$ [m²/kg]	μ [%]	m [%]	σ [%]	Training time [min]
3000	0.0378	0.0367	9.9	3.7	20.0	118



(a)



(b)

Figure 3.37 - Simulation results for $N_{obj,tr}$ equal to 3000. (a) RMSE of the training and validation phase. (b) True and predicted BC of the testing dataset.

It is important to highlight that the performance metrics assume quite different values for different ranges of *BC*. Three subsets of the initial *BC* interval have been identified:

- $0.022 \leq BC < 0.1$ m^2/kg
- $0.1 \leq BC < 1$ m^2/kg
- $1 \leq BC < 1.32$ m^2/kg

representing the 6.0%, 69.0% and 25.0% of the whole dataset respectively. In particular, as it can be seen in Table 3.27, the performance metrics for the medium and high *BC* intervals are one order of magnitude smaller than for the low *BC* interval. This is due to a limited observability of the *BC* for values smaller than 0.1 m^2/kg.

Table 3.27 – Mean, median and standard deviation of the absolute percentage error for three subintervals of the *BC*.

BC interval [m^2/kg]	μ [%]	m [%]	σ [%]
[0.022, 0.1]	67.8	62.9	37.0
(0.1, 1.0]	6.4	3.7	8.6
(1.0, 1.32]	3.0	2.6	2.1

The number of objects of the training set can affect the performance of the prediction model as mentioned in Section 3.2.2. To analyse the impact that the number of objects in the training set has on the performance of the neural network, such number has been increased to 5000 objects. The performance metrics are reported in Table 3.28. It can be seen that the performance gets worse. This means that the neural network undergoes overfitting.

Table 3.28 – Performance metrics for $N_{obj,tr}$ equal to 5000 objects.

$N_{obj,tr}$	RMSE [m^2/kg]	RMSE$_v$ [m^2/kg]	μ [%]	m [%]	σ [%]	Training time [min]
5000	0.0468	0.0434	12.4	3.4	41.7	173

Sensitivity to space weather drivers and atmospheric model uncertainty

For the sake of the analysis described in the previous sub-section, the training and the testing datasets have been generated using the exponential atmospheric model (EAM). This implies that the atmospheric density only varies as a function of the altitude. However, the density of the upper atmosphere changes because of complex interactions between the nature of the atmosphere's molecular structure, the incident solar flux, and geomagnetic (auroral) interactions. This leads to a strong dependence also on latitude, longitude, and space weather activity. Therefore, to capture such a

variability and assess whether the NN is able to learn a more complex model, the training/validation and testing datasets have been generated propagating the synthetic objects with the NRLMSISE-00 model (Picone et al., 2002). In particular, space weather activity is accounted for via two indexes: the planetary range index (K_p) or the planetary equivalent amplitude (A_p), and the observed 10.7-cm solar radio flux ($F_{10.7}$). In this section, three test cases are considered as described below. For each test case, the set of space weather coefficients has been used to generate both the training/validation and testing datasets.

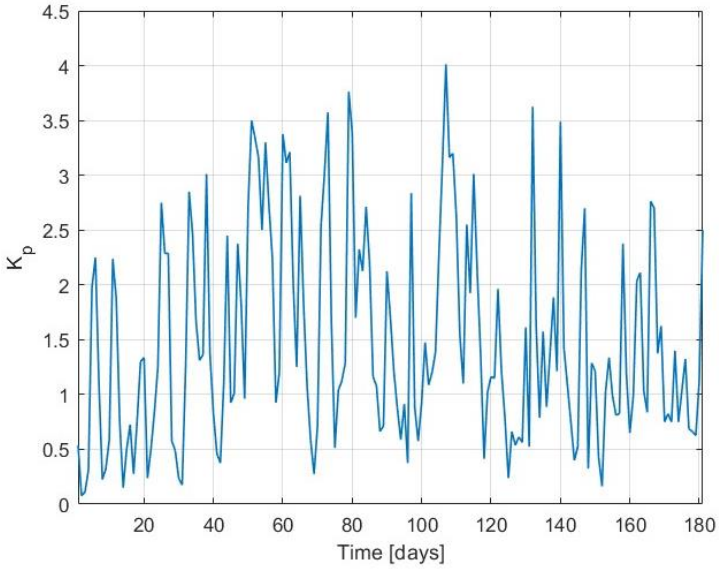
- Case A – Constant values of the space weather coefficients: K_p = 1, A_p = 4, $F_{10.7}$ = 80 SFU. This case is representative of low geomagnetic activity and solar flux. It is important for introducing the dependence on the geodetic coordinates without having large values of the atmospheric density.

- Case B – Constant values of K_p/A_p and time-varying $F_{10.7}$: K_p = 3, A_p = 15, $F_{10.7}$ computed according to [113]:

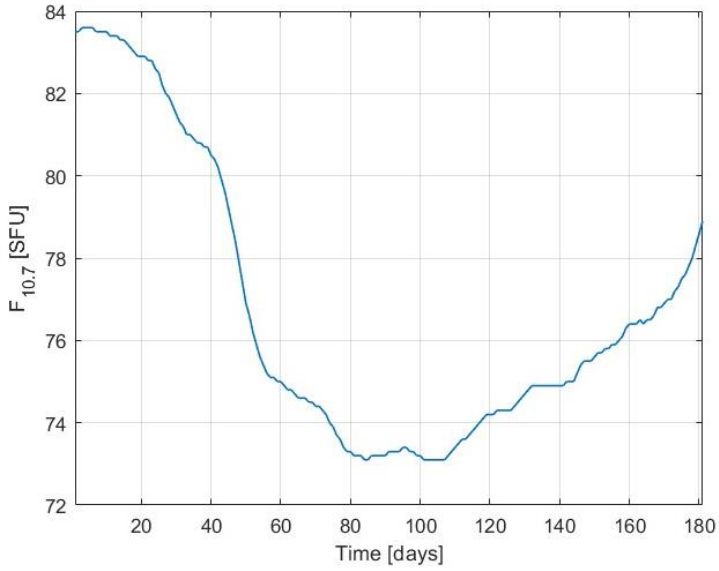
  $$F_{10.7} = 145 + 75\cos(0.001696\,t + 0.35\sin 0.00001695\,t)) \qquad (3.31)$$

  where t is the number of days from January 1, 1981. This case is representative of a stronger geomagnetic activity and solar flux, leading to larger orbital decays, thus increasing the number of re-entered space objects.

- Case C – Time-varying space weather coefficients. Updated values of K_p/A_p and $F_{10.7}$ are taken from [75] for the considered time. Figure 3.38 shows the daily mean K_p (a) and solar flux $F_{10.7}$ (b) for the considered elapsed time from the initial epoch 1st January 2021 at 00:00:00.000 UTC.



(a)



(b)

Figure 3.38 - Daily mean K_p (a) and $F_{10.7}$ (b) for the considered elapsed time from the initial epoch 1st January 2021 at 00:00:00.000 UTC.

Adding complexity to the atmospheric model while keeping constant the number of objects of the training dataset can lead to underfitting. This has been demonstrated by training the NN with 3000 and 5000 objects respectively and obtaining better performance in the latter case. Therefore, Table 3.29 reports the performance metrics

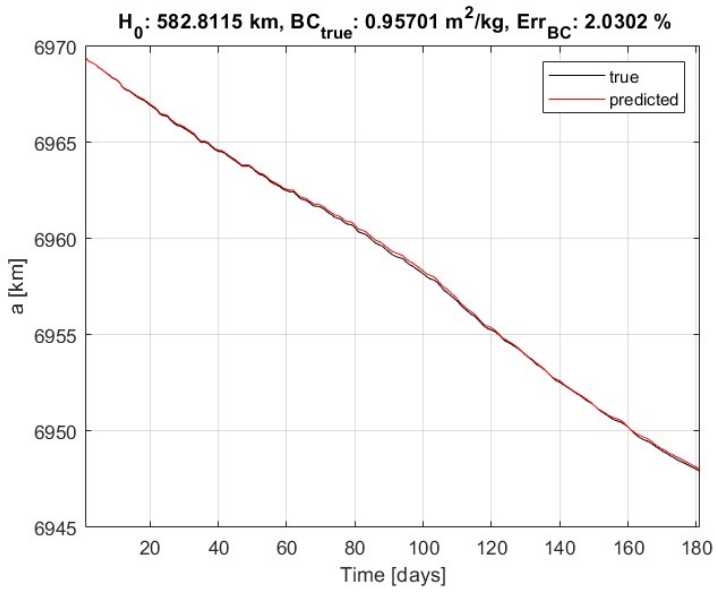
for the three test cases and $N_{obj,tr}$ = 5000. It can be noticed that the first and third test cases have similar performance. This is because, for the considered period of time (i.e., 1st of January 2021 to 30th June 2021), the mean values of the space weather coefficients used for Case C are similar to the constant values used for Case A. Therefore, the hourly and daily fluctuations of the coefficients are averaged out and have no significant impact on the phenomenon observability. For Case B instead, the performance metrics are slightly better than Case A and C, in particular in terms of RMSE and median. This is due to higher values of the space weather coefficients that, hence, improve the phenomenon observability leading to larger variations of semimajor axis and eccentricity. Moreover, it can be seen that adding complexity to the atmospheric model does not imply an increase in the computational cost of the training phase. The values are comparable to those shown in Table 3.28 (i.e., exponential model) with 5000 objects.

Table 3.29 – Performance metrics for test cases A, B and C with NRLMSISE-00 atmospheric model

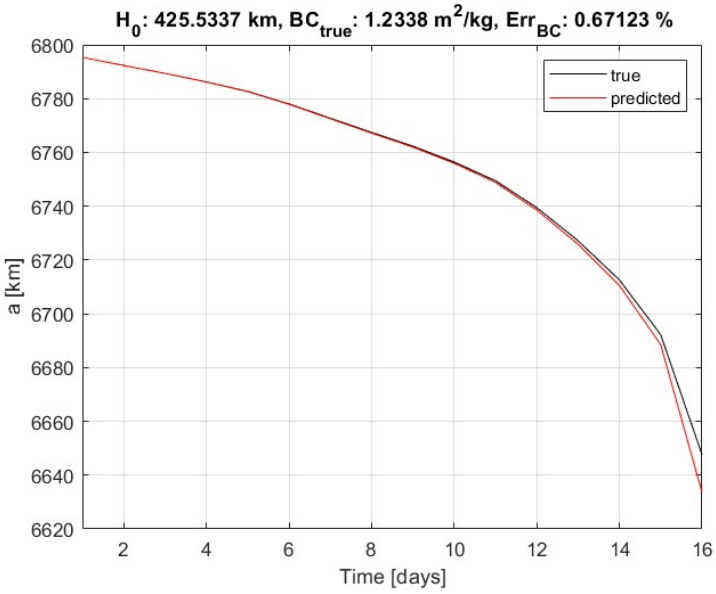
Test case	RMSE [m^2/kg]	$RMSE_v$ [m^2/kg]	μ [%]	m [%]	σ [%]	Training time [min]
A	0.0670	0.0694	9.9	6.7	11.2	164
B	0.0496	0.0510	10.3	5.4	17.9	158
C	0.0638	0.0640	10.4	6.5	13.2	170

For the last test case (C), the time evolution of the mean semimajor axis of four representative objects of the testing dataset is shown in Figure 3.39. In particular, it is compared the semimajor axis obtained propagating the RSO with the *BC* predicted by the neural network to the true semimajor axis. It can be noticed that, for an object with medium *BC* and a percentage error lower than 10% (Figure 3.39 (a)), the predicted semimajor axis follows the trend of the reference. If the *BC* is high and the initial altitude (H_0) relatively low, with a small percentage error (Figure 3.39 (b)), the predicted a follows the trend of the reference before starting to re-enter. If the object has a low *BC*, even with a percentage error larger than 15% (Figure 3.39 (c)), the error on the semimajor axis after 180 days is about 1 km. However, when the object has a medium *BC*, with a percentage error larger than 15% (Figure 3.39 (d)), the predicted a

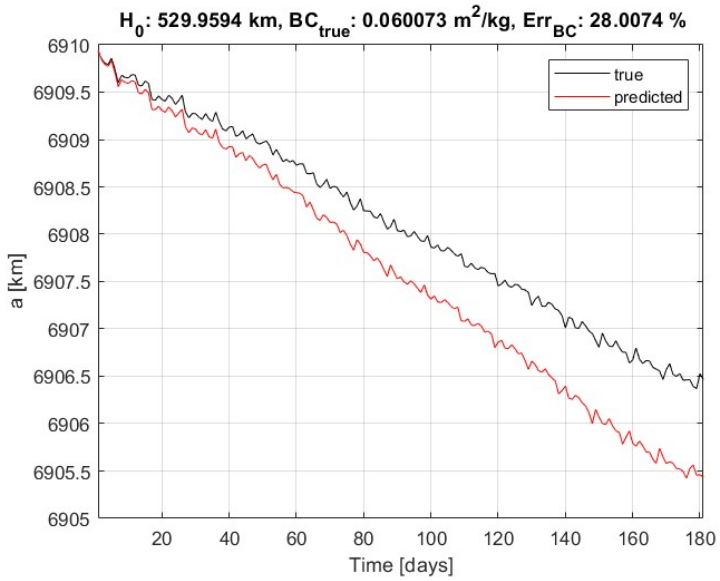
does not follow the trend of the true a and the error on the semimajor axis (Δa) after 180 days is about 15 km.



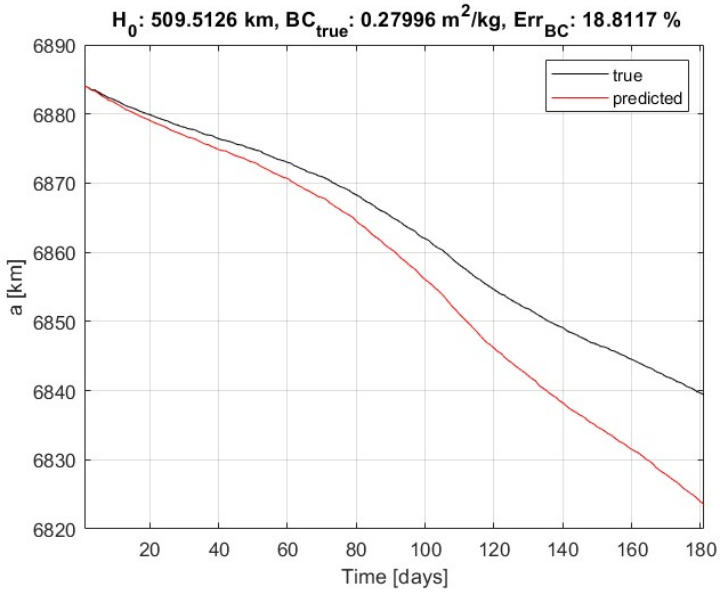
(a)



(b)



(c)



(d)

Figure 3.39 - Time evolution of the mean semimajor axis obtained with the *BC* predicted by the neural network and compared to the true semimajor axis, for four objects of the testing dataset. H_0 is the initial altitude of the RSO.

To study the impact of BC prediction errors on orbit propagation accuracy Figure 3.40 shows the empirical cumulative distribution function (CDF) of the absolute value

of Δa, varying the propagation time, for Δa < 5 km. After 1, 30 and 180 days, Δa is smaller than 1 km for 100%, 70% and 25% of the testing dataset, respectively. It must be highlighted that, to compute the CDF, only the space objects still on orbit (i.e., not re-entered) are considered.

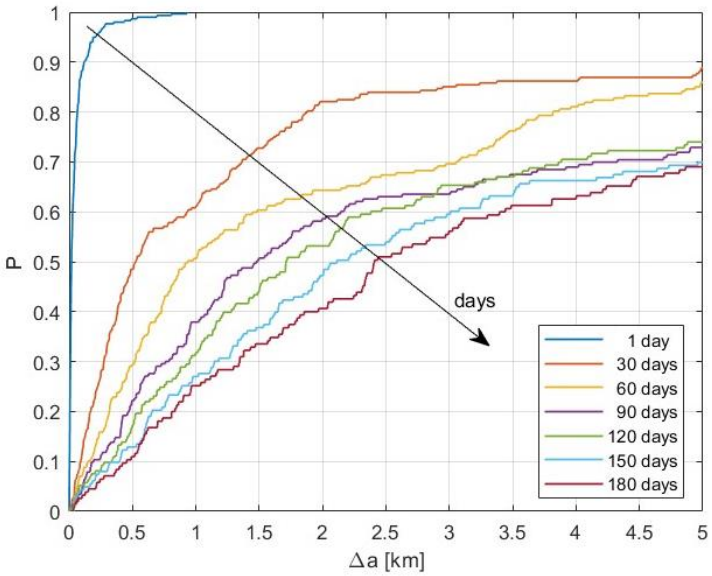
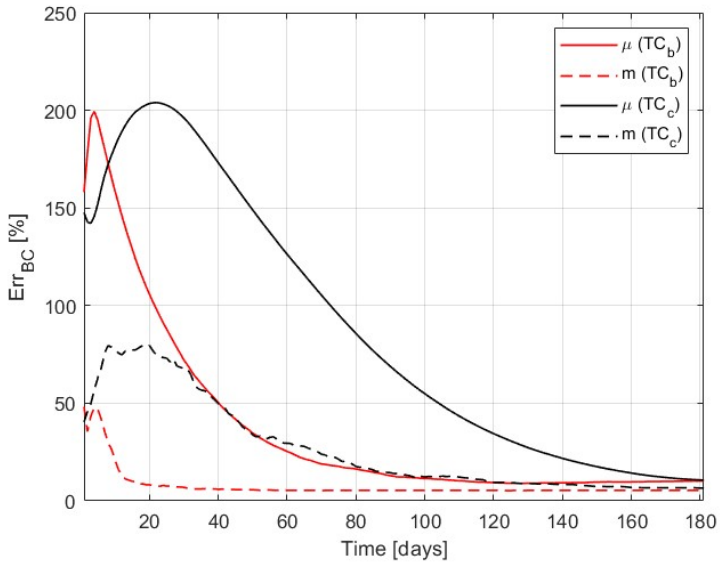


Figure 3.40 - Empirical CDF of the Δa varying the propagation time, for $\Delta a < 5\ km$.

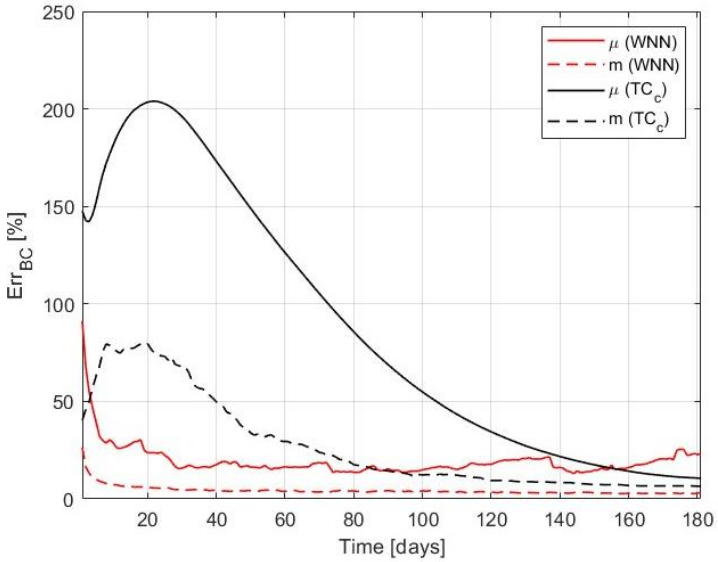
As demonstrated in Section 3.2.2, another important aspect is the accuracy of the prediction with respect to the propagation time. With this regard, for each RSO of the testing set, the *BC* has been predicted using the orbital elements within $n_{days} < N_{days}$ days, for n_{days} = 1, 2, …N_{days}. The value thus obtained is referred to as $BC_{pred,day}$. For every $BC_{pred,day}$, the absolute value of the percentage error is computed as in Eq. 3.6, hence having a distribution of Err_{BC} for each day.

Figure 3.41 (a) shows the comparison between the mean and median of this distribution for B and C. For a low solar activity (i.e., C), a longer propagation time is necessary to observe the trend of the orbital parameters and thus to reach the performance reported in Table 3.29. In particular, the median reaches the final value after almost 4 months. On the other hand, in the presence of a higher solar activity (i.e.,

B), for instance a geomagnetic storm, the median reaches the final value in about 20 days. In both cases, the significant difference between the mean and the median is due to the presence of a few objects (mostly the low *BC* interval) for which the relative error is large. Figure 3.41 (b) shows the comparison between the mean and median of the error distribution for C and the Wide Neural Network (WNN) analysed in Section 3.2.2. It can be noticed from the figure that the performance metrics are comparable after 4 months. However, the WNN reaches the final performance within 30 days. This might imply that the time evolution of the consistency errors has a clearer trend than the time evolution of the orbital parameters. It is also important to highlight that the WNN has been trained 10 times to consider the sensitivity of the performance with respect to the initial weights of the training phase. A single training, instead, has been carried out for the RNN proposed in this section. This can imply the presence of a few outliers and, thus, a larger value of the mean percentage error.



(a)



(b)

Figure 3.41 - Mean and median percentage error with respect to the propagation time: (a) Comparison between Case B and C; (b) Comparison between Case C and the Wide Neural Network of Section 3.2.2.

So far it has been assumed that the atmospheric density model used to generate the training and testing datasets is the same. In a real scenario, this would correspond to a very accurate knowledge of the actual atmospheric density. However, this is rather unlikely, due to the epistemic and aleatory uncertainties embedded in space weather and, hence, of the unpredictability of the actual atmospheric density values. To evaluate the robustness of the proposed algorithm to atmospheric model uncertainty, the following analysis has been carried out. The training dataset objects have been propagated with the NRLMSISE-00 model whereas the testing dataset objects with the EAM. As regards the SISE model, constant values of K_p, A_p, and $F_{10.7}$ equal to 6, 80, and 130 SFU have been chosen. This set has been chosen to assume that the mean value of the atmospheric density, at a specific altitude, as modelled with the SISE is comparable with the one estimated by the EAM. Nevertheless, locally (i.e., for a specific range of latitudes and longitudes) there can be a difference of the atmospheric density between the two models up to 100%. As an example, Figure 3.42 shows the percentage variation of the atmospheric density of the SISE model with respect to the

EAM, as a function of the latitude λ and the altitude h, for a fixed longitude ϕ equal to 0°, computed as:

$$\frac{\Delta\rho}{\rho} = \frac{\rho_{SISE} - \rho_{exp}}{\rho_{exp}} \, 100\% \tag{3.31}$$

where ρ_{SISE} and ρ_{exp} are the atmospheric densities computed with the SISE model and the EAM respectively. It can be noticed that, in this case, just by considering a different atmospheric model, up to 60% of variation of the atmospheric density can be obtained. Similar variations are obtained for different longitudes.

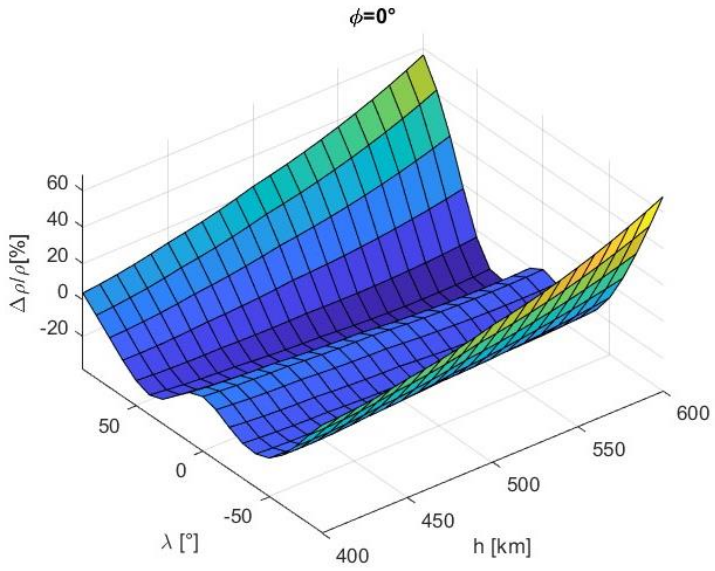


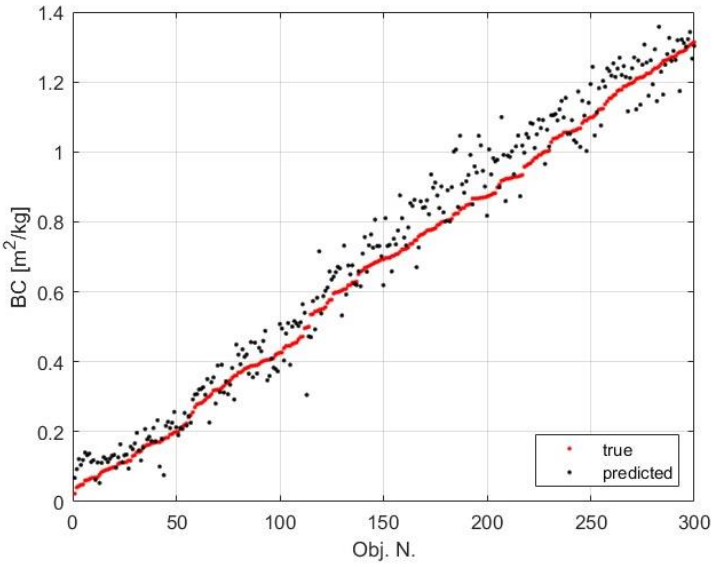
Figure 3.42 - Percentage variation of the atmospheric density of the NRLMSISE-00 model with respect to the EAM, as a function of latitude and altitude, for longitude equal to 0°.

Table 3.30 reports the performance metrics of this analysis, which differ by a few percentage points with respect to the previous test cases. However, the main outcome is the difference between the RMSE and the $RMSE_v$ which implies a worse generalization capability with respect to having the same atmospheric model for the training and testing datasets generation. Figure 3.43 (a) shows the predicted and the true *BCs* for each object of the testing dataset, while Figure 3.43 (b) shows the

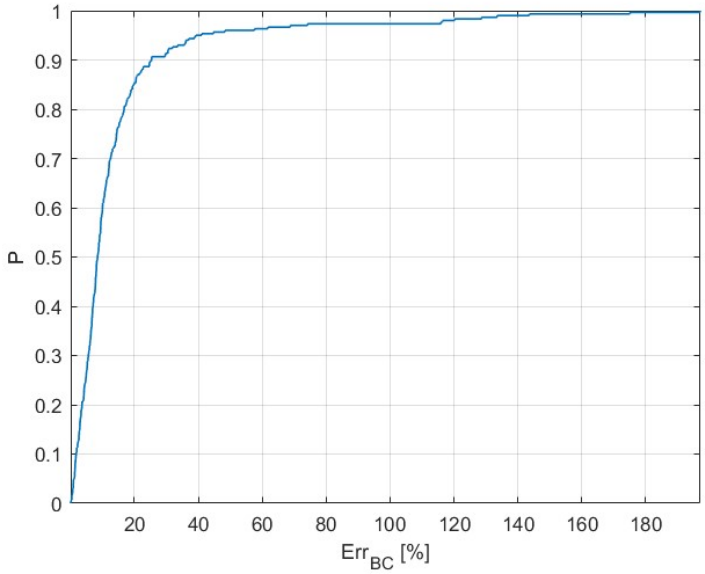
empirical Cumulative Distribution Function (CDF) of the percentage error. As expected, prediction performance worsens with respect to previous cases due to the atmospheric density being mismodelled. However, for 80% of the testing dataset the percentage error is smaller than 15%, and for 60% of the dataset it is smaller than 10%.

Table 3.30 – Performance metrics when considering different atmospheric models for the training and testing datasets.

RMSE [m^2/kg]	$RMSE_v$ [m^2/kg]	μ [%]	m [%]	σ [%]	Training time [min]
0.0673	0.0526	14.5	8.4	23.9	82



(a)



(b)

Figure 3.43 - Simulation results when considering different atmospheric models for the training and testing datasets. (a) True and predicted BC of the testing dataset. (b) Empirical CDF.

Variable measurement frequency

So far it has been assumed that the orbital elements of the training and testing datasets are provided at the same instants of time. In particular, starting from an initial epoch t_0, the i -th row of $\boldsymbol{X}$ corresponds to the i-th instant of time t_i:

$$t_i = t_0 + (i-1) * \Delta t, \qquad for\ i = 1, 2, \dots, N_{days} \tag{3.32}$$

where Δt is a fixed time interval, set equal to 1 day. For real applications, this is mostly unlikely. In fact, astrometric data might not be provided with a constant time interval and their update frequency varies depending on the orbital regime. To adapt the proposed algorithm to such circumstance and evaluate its performance, the following approach is adopted:

I. the synthetic objects of the testing dataset are propagated for 180 days as described in section 2,

II. a sampling mean frequency (SMF) is defined as number of measurements per day,

III. the orbital elements are thus sampled from the real trajectory with the defined SMF,

IV. the orbital parameters are interpolated on the same epoch of the training data set (i.e., 1 measurement per day as in Eq. 18)

Figure 3.44 shows the interpolation scheme with SMF equal to 2 measurements per day whereas Figure 3.45 shows the true, sampled and interpolated mean semimajor axis for a specific object of the testing dataset.

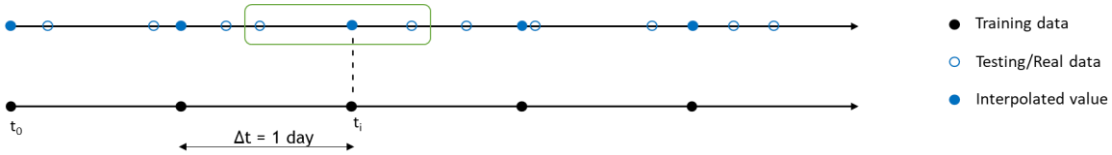


Figure 3.44 - Scheme of the interpolation process for SMF = 2 meas./day.

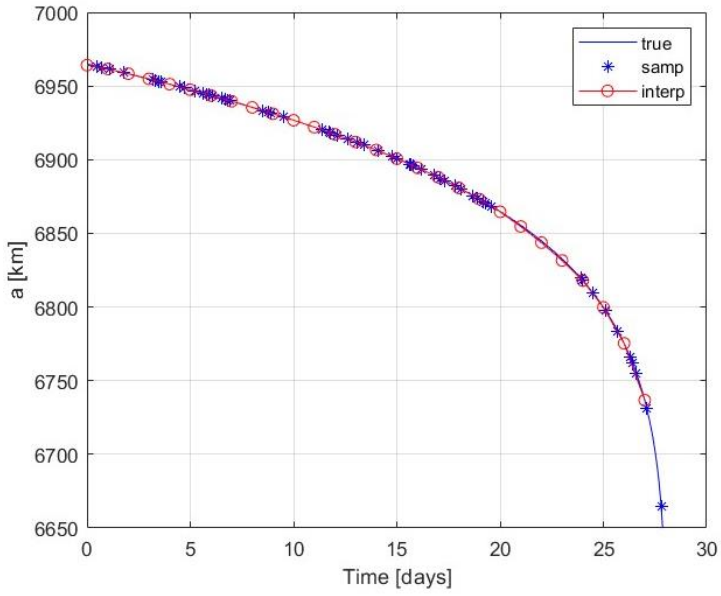


Figure 3.45 - True, sampled and interpolated mean semimajor axis for an object of the testing dataset.

Three different values of SMF equal to 1, 2, and 3 measurements per day have been considered. Table 3.31 reports the performance metrics for the three cases and for the

case in which no interpolation (NoInt) is needed (i.e., training and testing dataset are sampled on the same epochs). It can be noticed that the performance is still good for SMF = 1 meas./day and that, for SMF greater or equal to 2 meas./day, the performance metrics are comparable to NoInt.

Table 3.31 – Mean, median and standard deviation of the absolute percentage error for a sampling mean frequency equal to 1, 2 and 3 measurements per day. The last row reports the metrics when no interpolation is needed.

SMF [meas./day]	μ [%]	m [%]	σ [%]
1	13.5	8.3	17.2
2	10.6	7.2	16.2
3	10.6	6.7	16.5
NoInt	10.4	6.5	13.2

Robustness to measurement noise

In the results presented so far, it is assumed that the state vector of the RSO at each time instant is error-free; however, astrometric data will be affected by measurement noise due to limited sensor accuracy. To assess the sensitivity of the method with respect to orbit determination errors, the following analysis has been carried out.

Let $\boldsymbol{x}_i^j$ be the state vector of the j-th object of the testing dataset at the i-th time instant expressed in the RSW reference frame [136]. An uncertainty in the knowledge of position and velocity has been assumed to simulate a measurement error, thus leading to an estimated state $\boldsymbol{X}_{est,i}^j$:

$$\boldsymbol{X}_{est,i}^j = \boldsymbol{x}_i^j + \boldsymbol{\epsilon}_i^j \tag{3.33}$$

where $\boldsymbol{\epsilon}_i^j$ is the uncertainty. This uncertainty has been modelled as a white Gaussian noise on the objects' position and velocity in the RSW reference frame, with a standard deviation along the radial, along-track and cross-track directions as reported in Table

3.32. These values have been chosen considering typical orbit determination errors [113]. The estimated state thus obtained has then been converted into mean orbital parameters as described in Section 3.2.1

Table 3.32 – Standard deviation on position and velocity in the RSW reference frame.

	r_r [km]	r_s [km]	r_w [km]	v_r [m/s]	v_s [m/s]	v_w [m/s]
σ	0.2	1.0	0.3	0.1	1.0	0.1

In particular, to isolate the contribution of the position error from the velocity error (and viceversa), three different cases have been considered (it is assumed that no interpolation is needed):

I. Only position errors

II. Only velocity errors

III. Both position and velocity errors.

For each case, the performance metrics have been computed and reported in Table 3.33. It can be noticed that the algorithm is quite robust with respect to measurement noise in all cases. This is an important consideration especially when comparing the algorithm proposed in this section to that in Section 3.2.2, based on the consistency error. In fact, in Section3.2.2, an initial orbit determination error would be propagated to generate the *false* trajectory. This would lead, especially for low altitudes or/and high *BC*, to a consistency error which is significantly different to the error-free case, thus impacting on the overall performance.

For the sake of completeness, Table 3.33 shows also the case (IV) in which, besides position and velocity errors (i.e., case III), the testing set data are unevenly sampled and the SMF is equal to 2 meas./day. The performance slightly deteriorates, assuming values comparable to the previous section.

Table 3.33 – Mean, median and standard deviation of the absolute percentage error when including measurements noise on position (I), velocity (II) and both (III). Case IV considers also unevenly spaced data.

Case	μ [%]	m [%]	σ [%]
I	10.2	6.4	15.5
II	10.4	6.4	17.3
III	10.5	6.6	15.8
IV	11.0	7.0	17.0

4. Mission Characterization: Recognised Space Picture and Agile Manoeuvring Strategy

This chapter presents the tools developed in the RSP context and the results obtained with numerical simulations. Then, in the context of responsive space, an optimal manoeuvring strategy for overflying an Earth site is described as well as its application to simulated and real scenarios.

4.1 Recognized Space Picture

The term Recognized Space Picture (RSP) refers to a continuous representative service which provides the presentation of the following [137]:

- Detection, orbital tracking, classification (space surveillance) of artificial space objects
- Identification of space objects of interest (space reconnaissance)
- Ground and space threats assessment on space related activities, services and operations (threats assessment)
- Recognition, display and analysis of environmental phenomena (space weather), supported by a multi-layered multimedia information architecture.

In the space defense context, RSP designs the requirements of Space Forces for a surveillance function needed to carry out space operations, such as defensive/offensive counterspace or missile defense. In a way, RSP is the equivalent for the space domain of the Recognized Air Picture (RAP) commonly used in North Atlantic Treaty Organization (NATO) air control centres [138]. That is why, in this framework, tools and algorithms to support several RSP functions have been developed.

4.1.1 Footprint determination of spaceborne sensors

In this section, the algorithms developed to support overflight analysis capabilities are described. First, it is described the analytical model that provides the geodetic coordinates of the points discretizing the footprint of a spaceborne sensor, and the Cartesian coordinates of the points discretising Field of View (FOV), expressed in the Earth-Centred-Earth-Fixed (ECEF) reference frame. Depending on the payload type (i.e., radar, optical, telecommunications, GNSS), a FOV type is assigned (namely, pyramidal for radar and optical sensors, conical for telecommunications and GNSS satellites. Figure 4.1 shows the flow diagram of the developed algorithm.

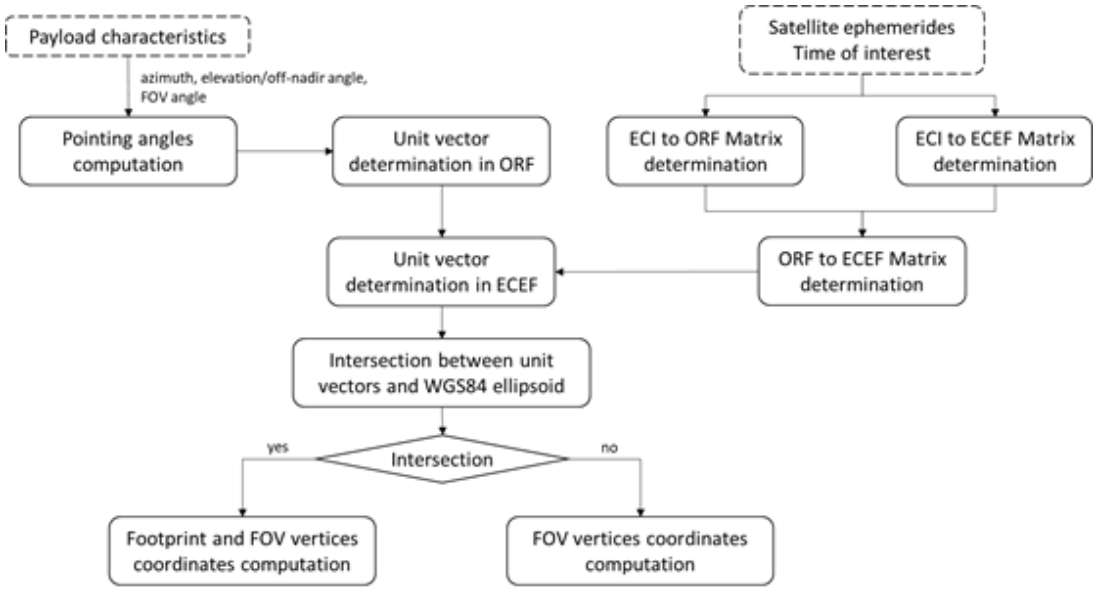


Figure 4.1 – Workflow of the footprint determination algorithm.

Considering the time of interest and the cartesian coordinates of the satellite in Earth-Centred-Inertial (ECI) reference frame, the ECI-to-ECEF, $\boldsymbol{M}_{ECI2ECEF}$, and ECI-to-ORF (Orbital Reference Frame), $\boldsymbol{M}_{ECI2ORF}$, rotation matrices are computed using, respectively, the SPICE library [139] and the following relations:

$$\boldsymbol{k}_{ORF} = -\frac{\boldsymbol{r}_{ECI}}{|\boldsymbol{r}_{ECI}|}, \quad \boldsymbol{j}_{ORF} = -\frac{\boldsymbol{r}_{ECI} \times \boldsymbol{v}_{ECI}}{|\boldsymbol{r}_{ECI} \times \boldsymbol{v}_{ECI}|}, \quad \boldsymbol{i}_{ORF} = \boldsymbol{j}_{ORF} \times \boldsymbol{k}_{ORF} \tag{4.1}$$

where $\boldsymbol{i}_{ORF}$, $\boldsymbol{j}_{ORF}$, and $\boldsymbol{k}_{ORF}$ are the unit vectors of the ORF reference frame; $\boldsymbol{r}_{ECI}$ and $\boldsymbol{v}_{ECI}$ are the position and the velocity vectors of the satellite in ECI. The ORF-to-ECEF matrix, $\boldsymbol{M}_{ORF2ECEF}$, is computed as:

$$\boldsymbol{M}_{ORF2ECEF} = \boldsymbol{M}_{ECI2ECEF}\boldsymbol{M}_{ECI2ORF}{}' \tag{4.2}$$

To determine the footprint, directions (in ORF) that discretize the FOV need to be identified. Each direction is defined by pointing angles, calculated differently based on the type of payload considered. For radar and optical sensors (i.e., pyramidal FOV), four directions are considered. Each direction $\boldsymbol{l}$ (solid red line in Figure 4.2) is defined by two rotations, one of type 3 (around the Z_{ORF} axis) and one of type 2 (around the Y_{ORF} axis) with angles:

- α: the angle between the projection of $\boldsymbol{l}$ onto the $(XY)_{ORF}$ plane (l_{xy}, dashed red line in Figure 4.2) and the X_{ORF} axis
- β: the angle between $\boldsymbol{l}$ and its projection onto the $(XY)_{ORF}$ plane.

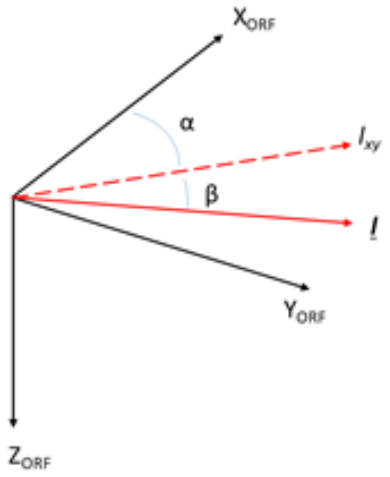


Figure 4.2 – Angles defining a direction $\boldsymbol{l}$ in the ORF.

In the case of radar, α takes values of $\pi/2 \pm 5°$, while β takes values of $-\pi/2 + \theta_1$ and $-\pi/2 + \theta_2$, with θ_1 and θ_2 as off-nadir angles. For optical sensors, α and β are calculated from FOV opening angles along the longitudinal (FOV_{el}) and transverse (FOV_{az}) directions, using the following relations:

$$\alpha = \frac{\pi}{2} \pm \text{atan}\left(\frac{tan\left(\frac{FOV_{el}}{2}\right)}{tan\left(\frac{FOV_{az}}{2}\right)}\right) \tag{4.3}$$

$$\beta = -\frac{\pi}{2} \pm \text{atan}\left(\sqrt{\left(tan^2\left(\frac{FOV_{el}}{2}\right) + tan^2\left(\frac{FOV_{az}}{2}\right)\right)}\right) \tag{4.4}$$

Once the direction vectors are obtained in ORF, they are converted to ECEF, obtaining the unit vector of components $\boldsymbol{i}_{ECEF}$, $\boldsymbol{j}_{ECEF}$, and $\boldsymbol{k}_{ECEF}$. For each direction is associated a line parametrically written as:

$$\begin{cases} X_{ECEF} = x_{ECEF} + \lambda * i_{ECEF} \\ Y_{ECEF} = y_{ECEF} + \lambda * j_{ECEF} \\ Z_{ECEF} = z_{ECEF} + \lambda * k_{ECEF} \end{cases} \tag{4.5}$$

which is intersected with the WGS84 ellipsoid [140], expressed in Cartesian form as:

$$\frac{X_{EFEC}^2 + Y_{ECEF}^2}{R_{eq}^2} + \frac{Z_{ECEF}^2}{R_p^2} = 1 \tag{4.6}$$

where X_{ECEF}, Y_{ECEF}, and Z_{ECEF} are the ECEF coordinates of the intersection point; x_{ECEF}, y_{ECEF}, and z_{ECEF} are the ECEF coordinates of the satellite; λ is a parameter; R_{eq} and R_p are the equatorial and polar radii associated with the WGS84 model. The intersection leads to solving a second-degree equation for λ. The equation will have two distinct real solutions if the line is secant (in this case, the smaller value of λ is considered), two coincident real solutions if the line is tangent, and no real solutions if there is no intersection. In the first two cases, λ allows calculating the ECEF coordinates of the intersection point, which can be subsequently converted into geodetic coordinates (i.e., latitude and longitude) [141]. The FOV can be built by

connecting the intersection points or, for a FOV that does not intersect the Earth's surface, considering a percentage (set at 90%) of λ and connecting the obtained vertices. In case of imaginary solutions, the FOV vertices can still be determined by assigning an arbitrary value to λ.

If the satellite is a telecommunication satellite (i.e., conical FOV), the developed tool can provide two outputs. The first one involves establishing a minimum ground elevation angle required for coverage and calculating the semi-opening angle of the cone centred on the satellite, assuming nadir pointing. Defining a minimum elevation angle $\alpha_{el,min}$ (with default value equal to 20°), the semi-opening angle is computed using the following equation:

$$\theta = \text{asin}\left(\frac{R\,cos\left(\gamma_{el,min}\right)}{R + h_{sat}}\right) \tag{4.7}$$

In this type of analysis, the circular footprint is defined by its centre and radius. The centre is determined as described above (e.g., as intersection between a line and the ellipsoid), considering the direction vector of components [0 0 1] in ORF. The base radius is calculated using geometric relations (see Figure 4.3):

$$\begin{aligned} c_1 &= a\cos(\theta) \\ c_2 &= c_1 - \sqrt{(c_1^2 - a^2 + R^2)} \\ \delta &= \text{asin}\left(sin(\theta)\frac{c_2}{R}\right) \end{aligned} \tag{4.8}$$

where a is the satellite's semi-major axis, R is the Earth's mean radius, θ is half of the FOV opening angle, and δ is the angle at the center as shown in Figure 4.3.

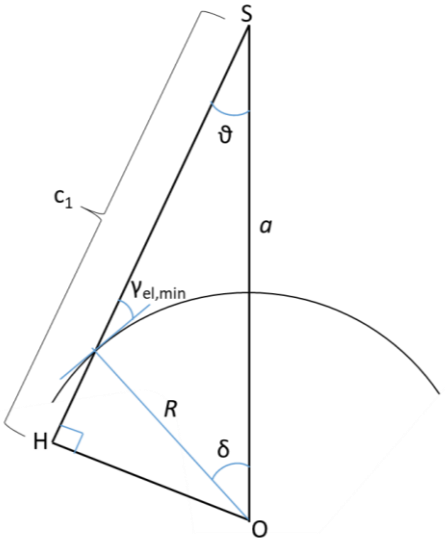


Figure 4.3 – Sketch of the geometry for the determination of the conic FOV radius.

The second output consists in determining the footprint in the case of a known opening angle and knowing the pointing angles relative to the cone axis. The direction of the cone axis $\boldsymbol{l}$ is defined in the ORF (as described in previous cases and shown in Figure 4.2), and a virtual plane (π) perpendicular to this direction is defined at a default distance of λ equal to 100 km. The ECEF coordinates of the intersection between the cone axis and the plane π are determined using Eq. 4.5 and 4.6 ($XYZ_{CE,OP}$), and the radius r_{OP} of the circumference obtained by intersecting the cone with the plane is calculated using the relation:

$$r_{OP} = \lambda \tan\left(\frac{FOV}{2}\right) \tag{4.9}$$

A local reference system is defined with the Z_{OP} axis perpendicular to π, pointing at the satellite, and the X_{OP} and Y_{OP} axes lying in the plane and orthogonal to Z_{OP}. The circumference is discretized into N points, each having the following coordinates in the local reference:

$$\begin{cases} X_{OP} = r_{OP} \cos\gamma \\ Y_{OP} = r_{OP}\, sin\, \gamma \\ Z_{OP} = 0 \end{cases} \tag{4.10}$$

with γ as the angle between the vector connecting the origin of the local reference to the generic point, and X_{OP}. At this point, the rotation matrix from the local reference to the ORF (M_{OP2ORF}) and the ECEF coordinates of the N points $(XYZ)_{ECEF}$ are calculated using the relations:

$$(XYZ)_{ECEF,OP} = M_{ECI2ECEF}\, M_{ORF2ECI}\, M_{OP2ORF}\, (XYZ)_{OP}$$
$$(XYZ)_{ECEF} = (XYZ)_{ECEF,OP} + (XYZ)_{CE,OP} \quad (4.11)$$

with $(XYZ)_{ECEF,OP}$ as the coordinates of the points in a shifted ECEF reference, having the origin coinciding with that of the local reference; $M_{ECI2ECEF}$ as the rotation matrix from ECI to ECEF; $M_{ORF2ECI}$ as the rotation matrix from ORF to ECI. Given the ECEF coordinates of the points and the satellite, it is possible to associate each point with the line connecting the satellite to the point and, consequently, the respective direction in ECEF and calculate the intersection with the ellipsoid according to Eq. 4.5-4.6.

In the case of GNSS satellites (i.e., conical FOV), the opening angle is calculated using Eq. 4.7 assuming that $\gamma_{el,min}$ is 5°. It is possible to discretize the footprint with canter and radius, or with the virtual plane method as described in the case of telecommunication satellites. In both cases, nadir pointing is assumed.

In the case of optical payloads, it is also necessary to estimate the AOI illumination conditions. For this purpose, the local solar time is computed, and if this value falls within a pre-defined interval, the area is assumed to have suitable illumination conditions.

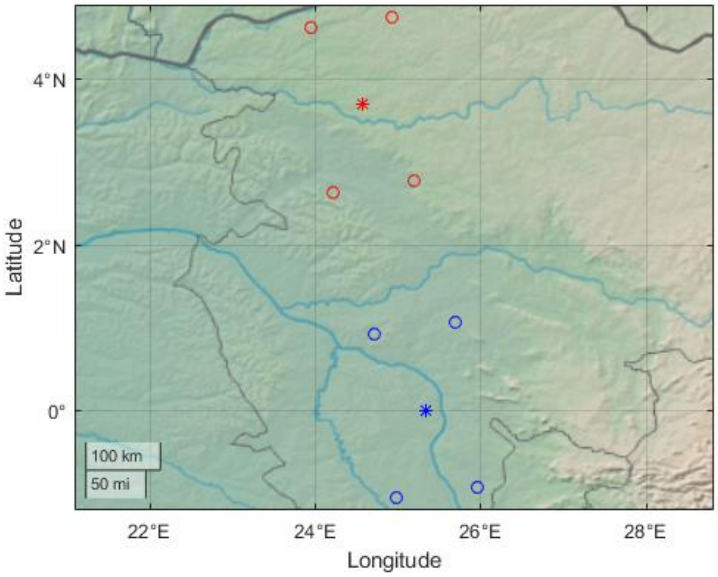
Performance assessment

Two test cases are presented to demonstrate the applicability of the proposed algorithms. In both cases, a reference satellite is considered that, at the beginning of the analysis, has the orbital parameters listed in Table 4.1.

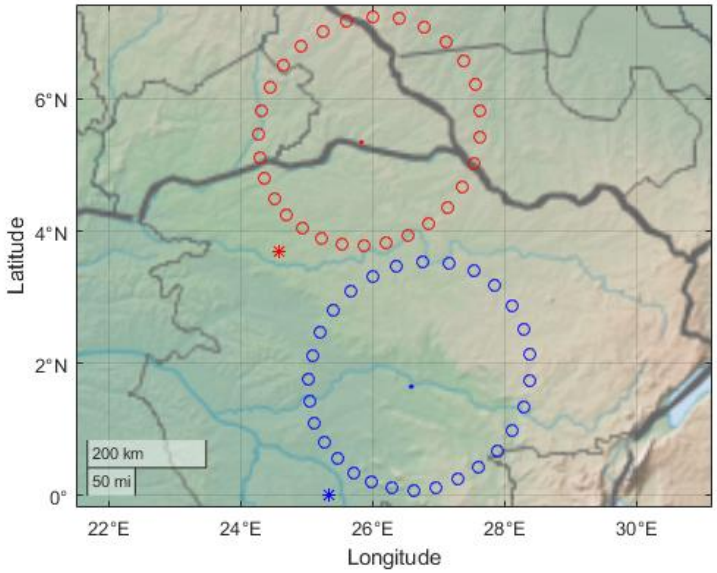
Table 4.1 – Initial orbital elements of the satellite.

$a\ [km]$	$e\ [-]$	$i\ [°]$	$\Omega\ [°]$	$\omega\ [°]$	$\nu\ [°]$
7006.96	0.00131	97.82	324.49	67.69	292.21

The analysis refers to two successive instants in time, the first corresponding to May 20, 2020, at 04:04:36.850 UTC, and the second to May 20, 2020, at 04:05:36.850 UTC. In the first test case, it is assumed that the satellite is equipped with an optical sensor with a FOV of 10° along the transverse direction and 20° along the longitudinal direction. The pointing is nadir. Figure 4.4 (a) shows the footprint generated at the two instants in time (blue and red circles), resulting from the intersection between the 4 directions corresponding to the sides of the pyramidal FOV and the Earth's surface. The projection of the satellite onto the Earth's surface (SSP) is indicated with an asterisk. In the second test case, it is assumed that the satellite is a telecommunication satellite. The field of view (conical) is rotated by angles $\alpha = 45°$ and $\beta = -70°$. Figure 4.4 (b) shows the corresponding footprint at the two instants in time (blue and red circles), the SSP (blue and red asterisks), and the intersection between the cone axis and the Earth's surface (blue and red points). It can be observed that, given the inclination of the cone, the points discretizing the footprint are denser near the SSP and become more scattered at a greater distance.



(a)



(b)

Figure 4.4 – Footprint for optical (a) and telecommunication (b) sensor.

4.1.2 Visibility analysis of a satellite from a ground-based sensor

The algorithm described in this section analyses whether a satellite is visible from a ground-based sensor. To this aim, it first models analytically the FOV of the ground-based sensor. Figure 4.5 describes the general architecture of the algorithm.

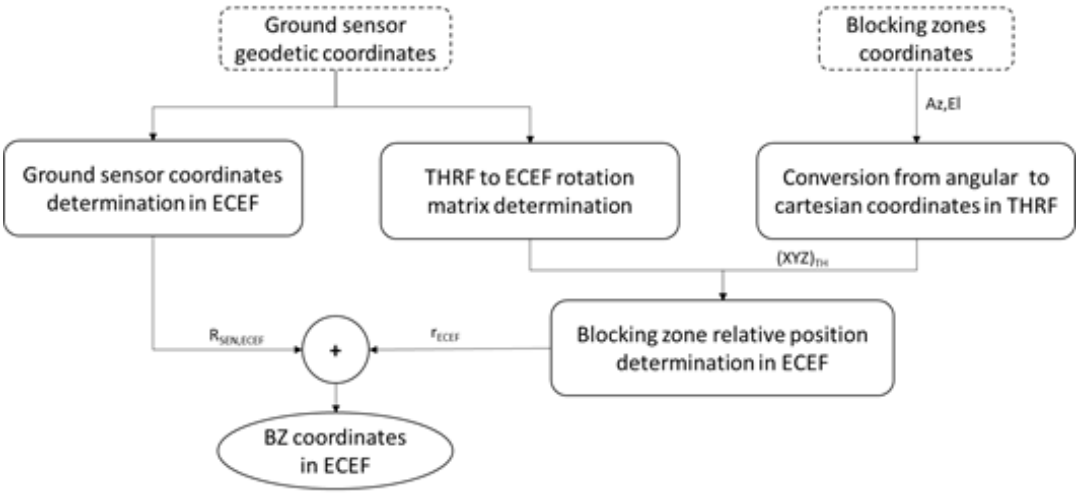


Figure 4.5 – Workflow of the Blocking Zones modelling algorithm.

A set of Blocking Zones (BZ) is defined in terms of azimuth and elevation intervals. Starting from the angular coordinates of points delimiting the BZ, the cartesian coordinates in the THRF (Topocentric Horizon Reference Frame) of these points are calculated using the relationships [141]:

$$\begin{cases} X_{TH} = \lambda_{FOV} \cos\beta \ \cos(\alpha) \\ Y_{TH} = \lambda_{FOV} \cos\beta \ \sin(\alpha) \\ Z_{TH} = \lambda_{FOV} \sin(\beta) \end{cases} \tag{4.12}$$

where β is the elevation angle and α is given by:

$$\begin{cases} \alpha = \pi - Az, \ \ se\ 0 \leq Az \leq \pi \\ \alpha = 3\pi - Az, \ \ se\ \pi < Az < 2\pi \end{cases} \tag{4.13}$$

where Az is the azimuth of the considered point. These coordinates are then converted from THRF to ECEF using the corresponding rotation matrix, obtaining the $\boldsymbol{r}_{BZ,ECEF}$

vector, which is the position vector of the considered point with respect to the ground-based sensor, expressed in ECEF. The geodetic coordinates of the ground sensor can be converted into ECEF coordinates, $\boldsymbol{R}_{SEN,ECEF}$, using the SPICE library [139]. Finally, the cartesian coordinates, in ECEF, of the points defining the blocking zones are computed as:

$$\boldsymbol{r}_{BZ,ECEF} = \boldsymbol{R}_{SEN,ECEF} + \boldsymbol{r}_{BZ,ECEF} \tag{4.14}$$

Once the FOV of the ground sensors is modelled, it is possible to verify if a satellite enters the FOV of the sensor. Figure 4.6 shows the flow diagram of the visibility analysis algorithm described below. In this figure, the non-visibility condition is depicted by a red cross.

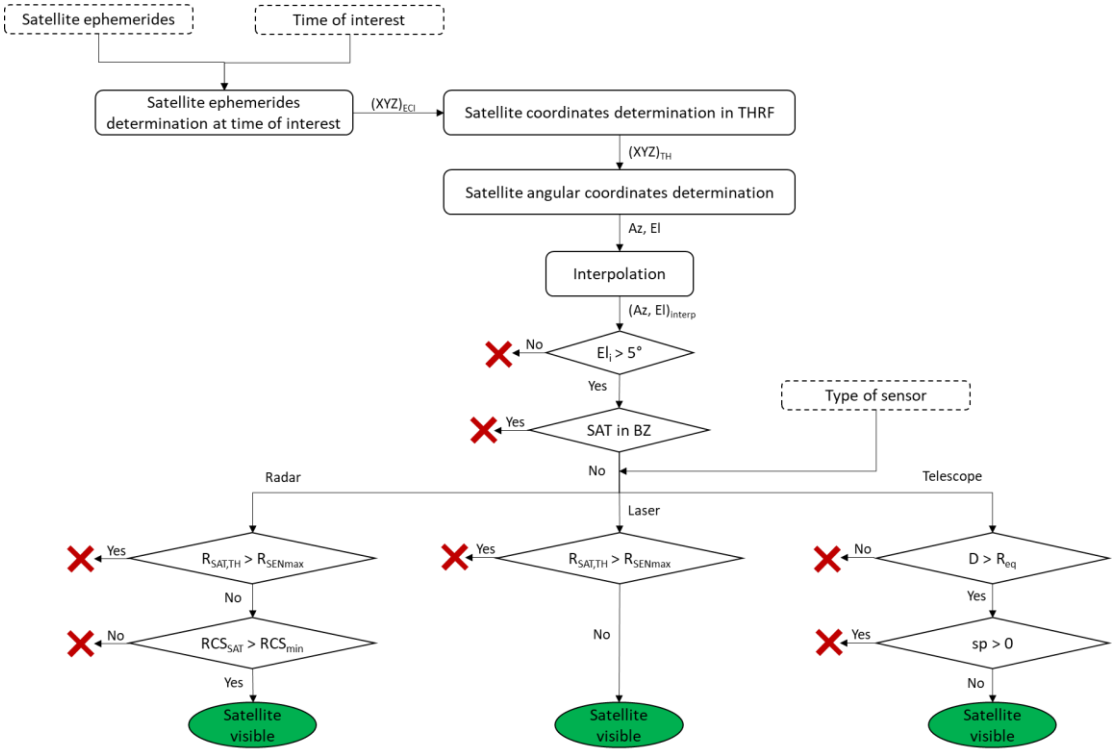


Figure 4.6 – Workflow of the visibility analysis algorithm.

The position of the satellite extracted from its ephemeris, and expressed in cartesian coordinates in the ECI reference frame, is expressed first in the THRF, $(XYZ)_{TH}$ [141],

and then converted into azimuth Az and elevation El in the THRF, using the relationships:

$$\begin{cases} Az = 180 - \text{atan}\left(\dfrac{Y_{TH}}{X_{TH}}\right) \\ El = \text{atan}\left(\dfrac{Z_{TH}}{\sqrt{X_{TH}^2 + Y_{TH}^2}}\right) \end{cases} \tag{4.15}$$

The visibility analysis consists of 3 subsequent filters:

1. If the satellite elevation is less than a threshold (set to 5°), the satellite is not visible. Otherwise, it proceeds to the next phase.

2. It is verified whether the satellite is in one of the blocking zones by comparing its angular coordinates with those defining the BZ. If the satellite is in the BZ, the satellite is not visible; otherwise, it proceeds to the next phase.

3. The third phase applies visibility filters based on the type of ground sensor considered. If the sensor is a radar, the range of the satellite in the THRF (r_{SAT}) is compared with the maximum range detectable from the sensor r_{max}; if $r_{SAT} > r_{max}$, the satellite is not visible. Otherwise, the Radar Cross Section (RCS) of the satellite (RCS_{SAT}) is compared with the minimum detectable RCS by the sensor (RCS_{min}); if $RCS_{SAT} > RCS_{min}$, the satellite is visible. Similarly, in the case of a laser, the range of the satellite in the THRF (r_{SAT}) is compared with the maximum range of the sensor (r_{max}); if $r_{SAT} > r_{max}$, the satellite is visible. If the sensor is optical, lighting conditions are considered. The positions of the Sun and the satellite in ECEF are calculated along with the quantity D, (*Dist* in Figure 4.7 [113]), using the relationships:

$$D = \left|R_{SAT,ECEF}\right| \cos\left(\zeta - \frac{\pi}{2}\right) \tag{4.16}$$

where ζ is the angle between the vector Earth-Sun position ($R_{SUN,ECEF}$) and the vector Earth-satellite position. If $D < R_{eq}$, the satellite is not illuminated, and thus not visible. If $D > R_{eq}$, it is evaluated whether the sensor is in shadow or illuminated: the dot product between $R_{SUN,ECEF}$ (Earth-Sun position vector)

and the position vector of the ground sensor in ECEF is calculated. If this product is positive, the ground sensor is illuminated by the Sun, and hence, the satellite is not visible. The satellite is visible otherwise.

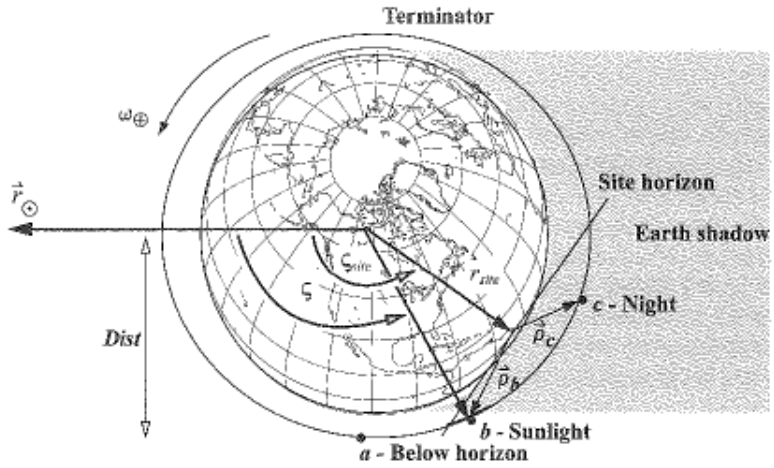
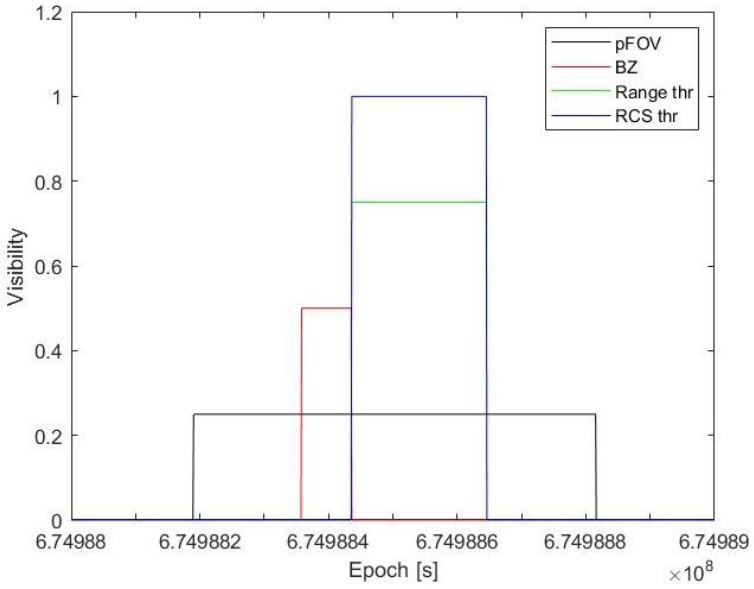


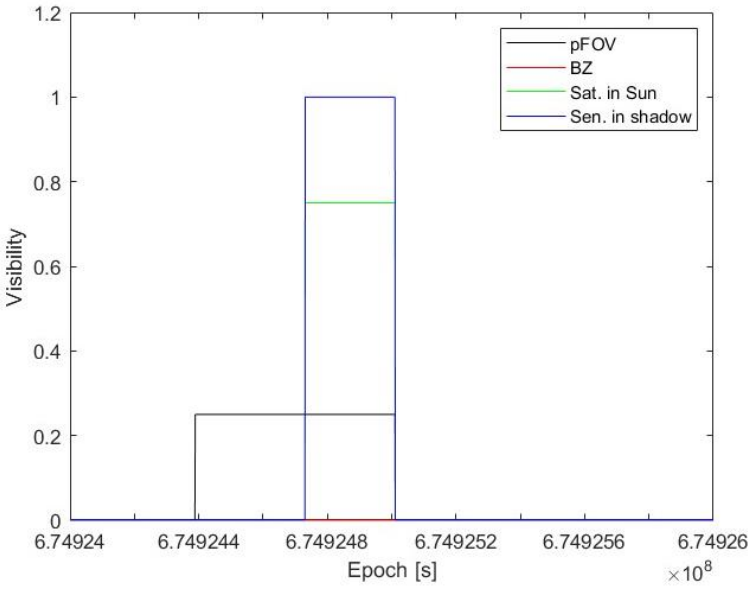
Figure 4.7 – Sketch of the Earth-Sun-Satellite relative geometry [113].

Performance assessment

Two test cases are presented in this paragraph, both related to Sentinel-3A. The first test case refers to a radar sensor located at latitude 37°, longitude -5°, and altitude 145 m; the second test case refers to a telescope located at latitude -33°, longitude 115°, and altitude 145 m. Considering a time interval of one day, the sensor detects the satellite multiple times. Figure 4.8 shows the visibility diagram for a single detection. It can be observed that for both the radar (Figure 4.8 (a)) and the telescope (Figure 4.8 (b)), various filters are applied: a) the satellite is in the potential field of view ($pFOV$) of the sensor (black line); b) the satellite is not in any blocking zone (red line); c) in the case of a radar sensor: the altitude of the satellite is below the maximum range of the sensor (green line), and its radar cross-section (RCS) is greater than the minimum detectable value by the sensor (blue line); in the case of a telescope: the satellite is illuminated by the Sun (green line), and the ground sensor is in shadow (blue line).



(a)



(b)

Figure 4.8 – Visibility diagrams for a radar sensor (a) and telescope (b).

4.1.3 GNSS coverage analysis

The algorithm described in this section enables the estimation of Vertical (VDOP), Horizontal (HDOP), Time (TDOP), Position (PDOP), and Geometric (GDOP) Dilution of Precision relative to one or more GNSS fleets and a selected Area of Interest (AOI). The analysis is based on the geometry of GNSS coverage and refers to a pre-defined time interval. Figure 4.9 illustrates the general architecture and logical flow of the algorithm.

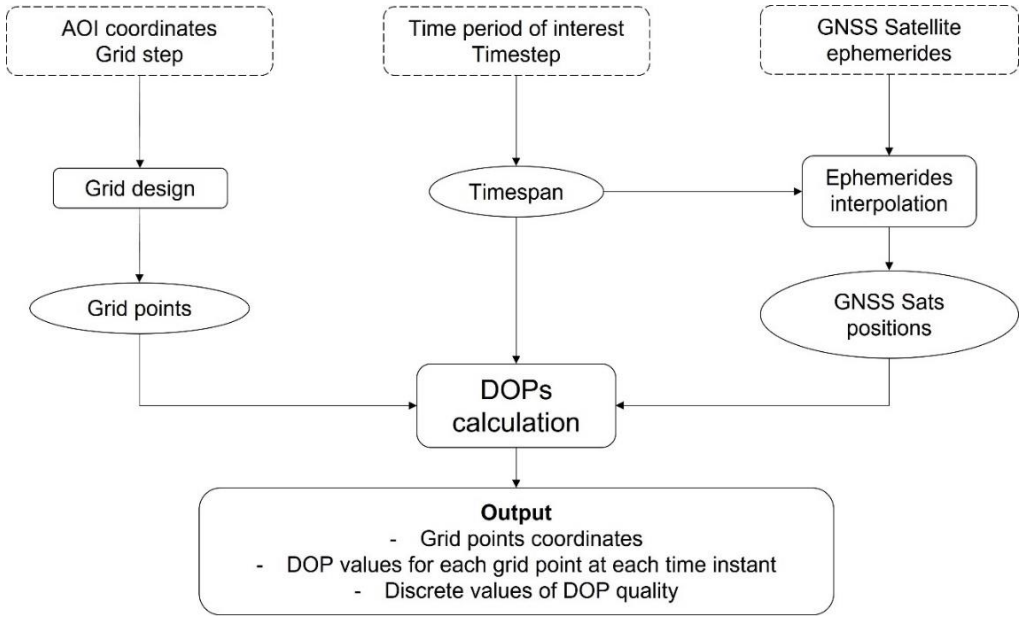


Figure 4.9 – Workflow of the GNSS coverage analysis algorithm.

The algorithm is described below, presenting the various steps chronologically:

1. The first step is to extract from the OEM files only the ephemerides within the time interval of interest for the analysis and interpolate them over the timespan obtained by discretizing the interval with the input timestep. This provides the positions of all GNSS satellites under analysis at each moment of interest.

2. The second step involves creating a grid of points over the provided AOI. First, the latitude step size ϕ_{step}, expressed in degrees, is calculated as constant,

$$\phi_{step} = \frac{l_m \cdot 180°}{L} \tag{4.17}$$

where l_m is the grid step provided as input (in km), and L is the length of the meridian on the WGS84 ellipsoid, equal to 20003.93 km. From this, the latitude at each step within the range of interest is derived. Subsequently, at each latitude step, the longitude step size is calculated as follows,

$$\lambda_{step,i} = \frac{l_m \cdot 360°}{(2\pi R_i \cos(\phi_i))} \tag{4.18}$$

for $i = 1,\ldots,N$ (N number of latitude steps), where ϕ_i is the i-th latitude within the interval, and R_i is the Earth's radius corresponding to that latitude, calculated as,

$$R_i = \sqrt{\frac{\left({R_{eq}}^2 \cos(LAT_i)\right)^2 + \left({R_p}^2 \sin(LAT_i)\right)^2}{\left(R_{eq} \cos(LAT_i)\right)^2 + \left(R_p \sin(LAT_i)\right)^2}} \tag{4.19}$$

where R_{eq} is the equatorial radius of the Earth, and R_p is the polar radius. At each latitude step, the longitude steps within the range of interest are then computed. Each latitude-longitude pair represents a point on the grid. The resulting $[2xM]$ matrix contains latitude and longitude values for all points forming the grid, where M is the number of points.

3. At this point, for each grid point, DOP values are calculated at each time instant, as described below. For each grid point, a North-East-Down (NED) coordinate system [142] is established with the origin at that point, and the positions of GNSS satellites are transformed from ECI to NED. The elevation angle is then computed for each satellite in the NED reference, and if it exceeds the mask angle, the satellite is considered in view. If there are at least 4 satellites in view, the DOP calculation proceeds. As an example, if exactly four

satellites are in view, denoting $\hat{x}_i$, $\hat{y}_i$, and $\hat{z}_i$ as the unit vectors from the grid point to the i-th satellite in the NED reference, the matrix H is built,

$$H = \begin{bmatrix} \hat{x}_1 & \hat{y}_1 & \hat{z}_1 & 1 \\ \hat{x}_2 & \hat{y}_2 & \hat{z}_2 & 1 \\ \hat{x}_3 & \hat{y}_3 & \hat{z}_3 & 1 \\ \hat{x}_4 & \hat{y}_4 & \hat{z}_4 & 1 \end{bmatrix} \tag{4.20}$$

and the covariance matrix Q is formulated as follows,

$$Q = (H^T H)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{xt} & \sigma_{yt} & \sigma_{zt} & \sigma_t^2 \end{bmatrix} \tag{4.21}$$

From here, the DOP values are obtained as:

$$\begin{aligned} VDOP &= \sqrt{\sigma_z^2} \\ HDOP &= \sqrt{\sigma_x^2 + \sigma_y^2} \\ TDOP &= \sqrt{\sigma_t^2} \\ PDOP &= \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \\ GDOP &= \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_t^2} \end{aligned} \tag{4.22}$$

4. Finally, the algorithm ends with a post-processing phase, assigning discrete levels to the DOP values based on certain thresholds. The default and recommended thresholds are summarized in Table 4.2.

Table 4.2 – DOP values rating.

DOP value	Rating
< 1	Ideal
1-2	Excellent
2-5	Good
5-10	Moderate
10-20	Fair
> 20	Poor

Performance assessment

To demonstrate the applicability of the algorithm, a test case is presented. Considering an Area of Interest (AOI) with the following vertices (latitude, longitude):

$$(40°, -120°), (40°, -90°), (55°, -90°), (55°, -120°)$$

a grid of evenly spaced points with a step of 100 km is built. The grid is shown in Figure 4.10 and consists of 390 points. Only the GPS constellation was considered for this case, for an analysis time of three days, from October 10, 2021, at 12:00:00 UTC to October 13, 2021, at 12:00:00 UTC. Figure 4.11 depicts the time evolution of TDOP, PDOP, and GDOP for a grid point located at 46.3° latitude and -114° longitude. It can be observed that, for most of the analysis time, the DOP values are below 5, corresponding to a rating of at least "Good" according to Table 4.2.

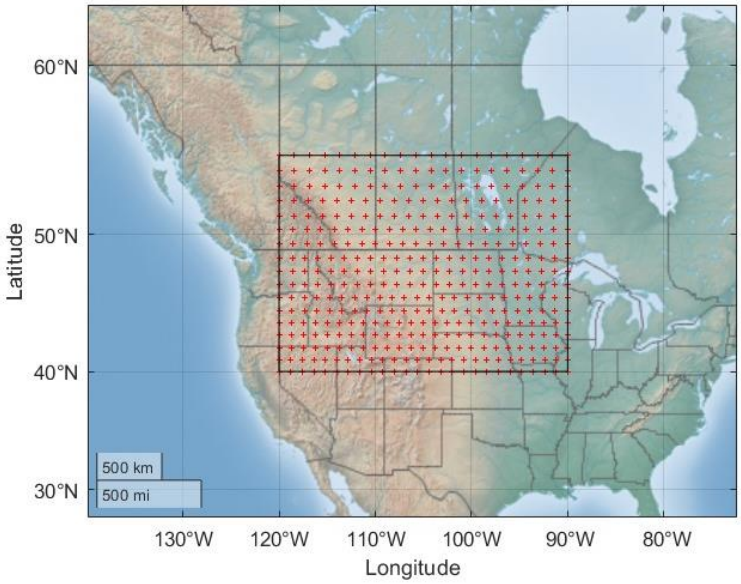


Figure 4.10 – Example of a grid for the considered AOI.

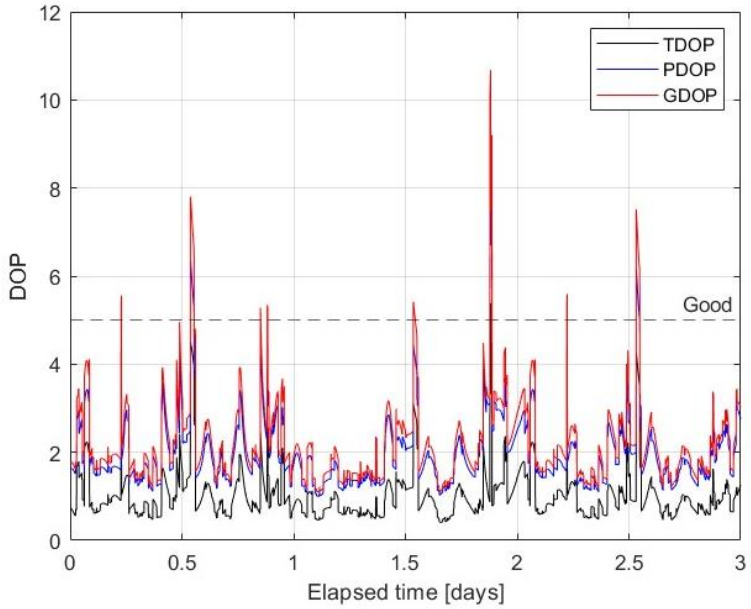


Figure 4.11 – Time evolution of TDOP, PDOP and GDOP for the grid point of coordinates 46.3° and -114° of latitude and longitude respectively.

4.2 Responsive Space

4.2.1 Introduction

In recent years, the capability to respond to real-time events by providing data on-demand for time-critical tasks has become paramount, especially for monitoring or search and rescue operations during natural disasters like earthquakes, forest fires, and floods [143], [144]. To enhance space responsiveness, two different strategies can be adopted. On the one hand, miniaturization and standardization of space systems can be exploited to enable responsive launch capabilities, after solving an optimal orbit design problem [145], [146], [147]. On the other hand, orbital and/or attitude manoeuvres can be commanded to adjust the ground tracks of in-orbit satellites, thus achieving fast Earth observation for specified ground targets [148]. The latter approach, referred to as the ground-track adjustment, has attracted increasing interest in the last few years.

The ground-track adjustment problem has been addressed in current literature by using either impulse manoeuvres (i.e., based on chemical propulsion systems) or low-thrust manoeuvres (i.e., based on electric propulsion systems). Co et al. [149] presented a method to quantify the terrestrial distance between the sub-satellite points of a non-manoeuvring satellite in a reference orbit and a manoeuvring one using chemical or electric propulsion for a given propellant budget, initial orbit and available manoeuvring time. In order to design a suitable initial orbit, they also investigated how the orbital elements affect terrestrial distance, and they found that altitude and eccentricity have the greatest effects [149]. Zhang and Cao [150] derived a single-impulse method to reach a free (i.e., with no constraints) elliptical final orbit and a shape-based low-thrust method to get an assigned final orbit (e.g., repeated ground track orbit or sun synchronic orbit) to solve the coplanar ground-track-adjustment problem. For the single-impulse method, they obtained the semi-major axis of the final orbit by solving a cubic equation. For the continuous low-thrust case, instead, the desired trajectory is modelled as a specified-form shape function with coefficients

obtained from boundary conditions. Both methods are based on the time difference for the two-body model, to avoid dealing with the J_2 perturbation, which is only considered in computing the longitude difference [150]. For the cases in which impulsive propulsion is available, Zhang et al. [151] proposed a one- and two-impulse approach to reach an elliptical and a circular free final orbit, respectively. In particular, approximate analytical solutions are derived by solving analytically the semi-major axis of the final orbit accounting for the linear J_2 perturbation, considering the possibility of either reaching an exact overflight on a ground target or keeping it with a conical sensor field of view. In contrast to the impulsive approaches suggested in [150] and [151], which achieve free final orbits whose ground-track passes over the Earth site, Zhang and Sheng [152] addressed the impulsive coplanar ground-track adjustment problem to get an assigned final orbit. Specifically, they introduced three-impulse and four-impulse methods for coplanar circular and noncoaxial elliptical orbits, respectively. Recently, Sheng and Geng [144] provided an analytical approach based on the normalized nodal distance to solve the J2-perturbed ground-track adjustment problem under apsidal-altitude constraints using a single impulse.

The above-mentioned ground-track adjustment methods focused on finding manoeuvring solutions for observing one target. However, the ground-track can also be adjusted to observe multiple Earth sites. For the low-thrust manoeuvre case, Co and Black [153] developed a control algorithm for a single low Earth orbit (LEO) satellite equipped with an electric propulsion system to pass over 10 randomly selected ground targets. Guelman and Shiryaev [154] introduced a closed-loop control algorithm that allows the direct flyover of the selected Earth sites without the need of precomputed reference trajectories. For the impulsive manoeuvre case, Mok and Bang [155] proposed near-optimal solutions for overflying one, two and three ground targets. To this purpose, they considered at least one manoeuvre to observe each target. Lin et al. [148], instead, used a single coplanar impulse to solve the ground-track adjustment problem for overflying one to three ground targets at most.

Nowadays, the ground-track adjustment approach becomes particularly appealing for space actors (i.e., space agencies, industries and private stakeholders) which can rely on more than one satellite to manoeuvre. Indeed, the possibility to rely on multi-

satellite systems helps in optimizing the trade-off between propellant expense and responsive observation requirements. In this context, Yu et al. [156] suggested two constellation design approaches based on the reachable domain belts of a single satellite. Reachable domain belts are generated by making use of the ground-track shift due to the period difference between a reference orbit and a manoeuvred one. Two- and one-impulse manoeuvres are considered to this purpose. Other works, instead, dealt with the reconfiguration of a constellation of multiple satellites. De Weck et al. [157] used the auction algorithm to determine how to best assign each satellite of an existing constellation to a spot in a new constellation such that the fuel consumption required for the reconfiguration is minimized. McGrath and Macdonald [158] considered a constellation of 24 satellites in four orbital planes to be manoeuvred into repeating ground-track orbits, and introduced a simple utility function to decide which satellite should be manoeuvred during each reconfiguration. Their work demonstrated that reconfiguring a constellation of satellites can provide increased coverage of 1.6–10 times compared to a static, global-coverage constellation.

Given this framework, this PhD thesis proposes an original decision-making architecture for a responsive multi-satellite system in LEO. The architecture outputs an optimal solution to overfly a user-specified Earth site within an assigned time frame considering both manoeuvring and non-manoeuvring options. Within the architecture, the ground track adjustment problem is addressed building upon the analytical method proposed by Zhang et al. [151]: however, unlike previous works, the proposed methodology accounts for the availability of different types of remote sensing sensors, i.e., either optical cameras or radars, and it considers a pyramidal-shaped field of view, while the Earth site is defined as an area of interest (AOI) characterized by a range of latitudes and longitudes. Once all feasible options are identified, a multi-objective optimization problem is formulated weighting both fuel consumption and time responsiveness constraints. A numerical analysis on simulated test cases is carried out to show the effect on performance of variability in the type of on-board sensors, in the area of interest, and in the number of available satellites. Finally, the architecture is also tested considering a simulated scenario involving a set of real Earth observation satellites.

4.2.2 Methodology

A flow diagram of the decision-making architecture for multi-satellite responsive manoeuvring is shown in Figure 4.12 .

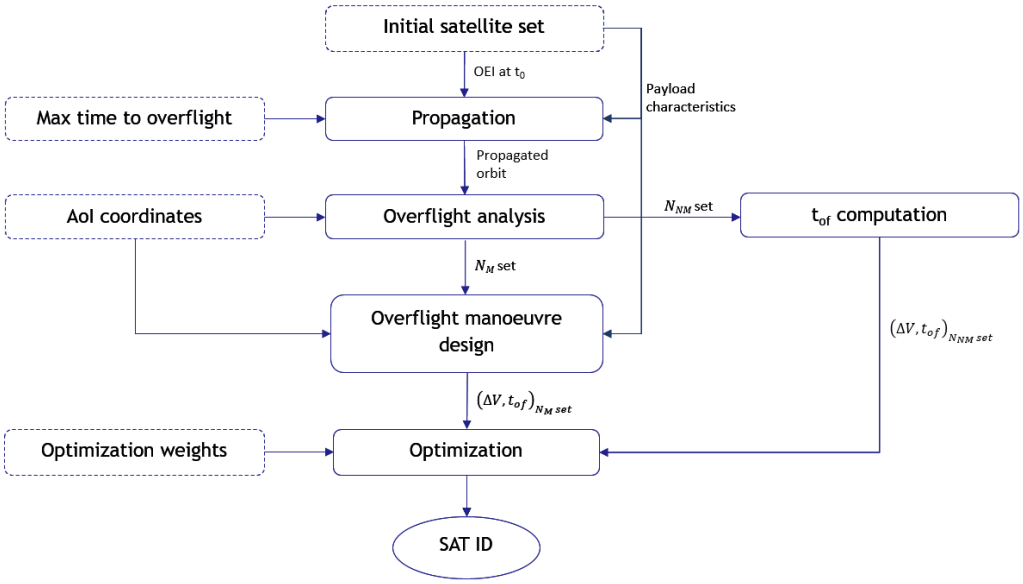


Figure 4.12 - Flow diagram of the decision-making architecture for multi-satellite responsive manoeuvring.

This architecture requires the inputs listed below.

1. A set of N_{sat} satellites, each identified by its orbital parameters (OEl_{0i}) at an initial epoch t_0 and by its physical features (i.e., mass, m, area exposed to the drag, A_{drag}, and drag coefficient, C_d). In addition, each satellite is assumed to be equipped with either an optical or radar Earth-observation sensor characterized by a pyramidal-shaped field of view (FOV). The optical sensor is assumed to be nadir pointing, and its FOV is described by the angular size in the cross-track (α) and tangential (β) directions; the radar sensor FOV is instead defined by the minimum, θ_{min}, and maximum, θ_{max} , off-nadir angles as well as the in-track

angle, γ. A graphical representation of the sensors' FOV modelling is provided in Figure 4.13.

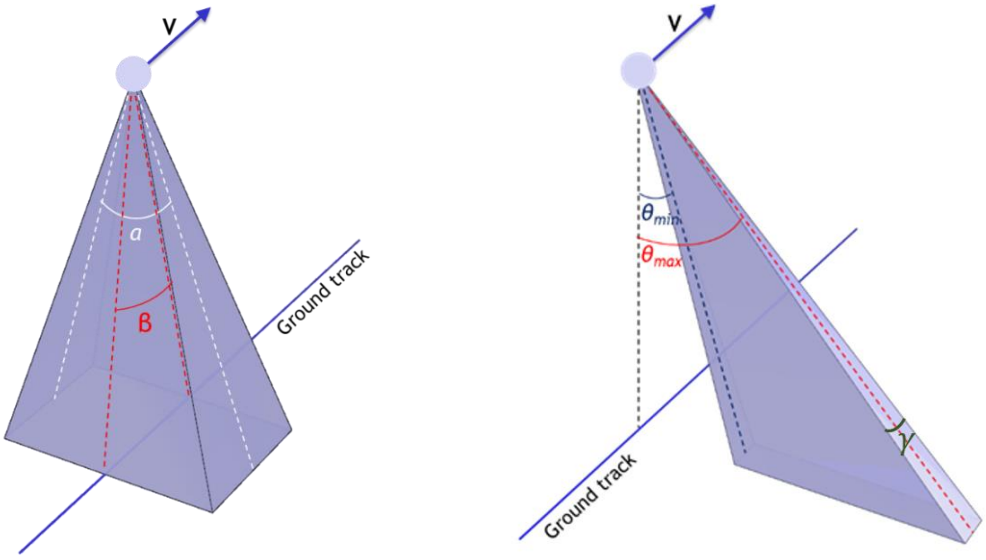


Figure 4.13 - Graphical representation of the optical and radar sensor Field of View and ground swath.

2. An Earth site, i.e., the target area of interest. This area is defined as a polygon, whose vertexes are assigned in terms of geodetic coordinates, namely latitude, ϕ, and longitude, λ.

3. The maximum admissible time to overflight, D, expressed as an integer number of days.

4. The weights for the optimization algorithm that measure the relative importance between the constraints on propellant consumption (expressed in the form of the total ΔV required for the responsive manoeuvre) and on the time to overflight, t_{of}. The definition and rational for selection of these weights is treated later on in this section.

In the propagation block, the trajectories of the N_{sat} satellites are propagated for D days starting from t_0 to check whether an overflight would occur within the given time

constraint without the need of manoeuvring. The propagation is performed by using NASA's General Mission Analysis Tool (GMAT), including the J_2 effect and drag, which are the most important perturbations in LEO.

The resulting ephemerides, the AOI coordinates, and the remote sensing payload's specifications are used as input of the overflight analysis block, which is applied to each satellite, according to the following steps. First, the footprint, i.e., the projection of the sensor FOV on the Earth surface must be computed. To this aim, the FOV is modelled as the solid angle identified by a rectangular pyramid with apex in the satellite position (assuming a nadir pointing and an off-nadir pointing for the optical and for the radar sensor, respectively). At each time instant of the ephemerides, the ground points delimiting the footprint are computed in geodetic coordinates as the intersections between the vertices of the pyramidal FOV and the spherical Earth. The overflight analysis can be made more time efficient by filtering out all the time intervals in which the AOI observation is not possible (e.g., the satellite is overflying the anti-meridian of the AOI). To this aim, a rectangular safe area is built considering the rectangle that includes the initial AOI and, then, increasing the dimensions of such a rectangle by values equal to the maximum variations of longitude and latitude between two consecutive time instants in the reference ephemerides, as shown in Figure 4.14. After that, the sensor's footprint is evaluated for each time instant of the reference ephemeris and it is checked whether any vertices of the footprint fall within the safe area. In this way, all time intervals where the AOI overflight is possible can be obtained. For each of these time intervals, the timespan is discretized with a user-defined time step (at least one order of magnitude smaller than the time step of the ephemerides), which is used to interpolate the position and velocity vectors of the ephemerides. At each time step of the identified time intervals, the sensor's footprint is evaluated to check whether at least one of the vertices of the footprint falls within the AOI or alternatively whether the sides of the footprint polygon intersect the sides of the AOI polygon.

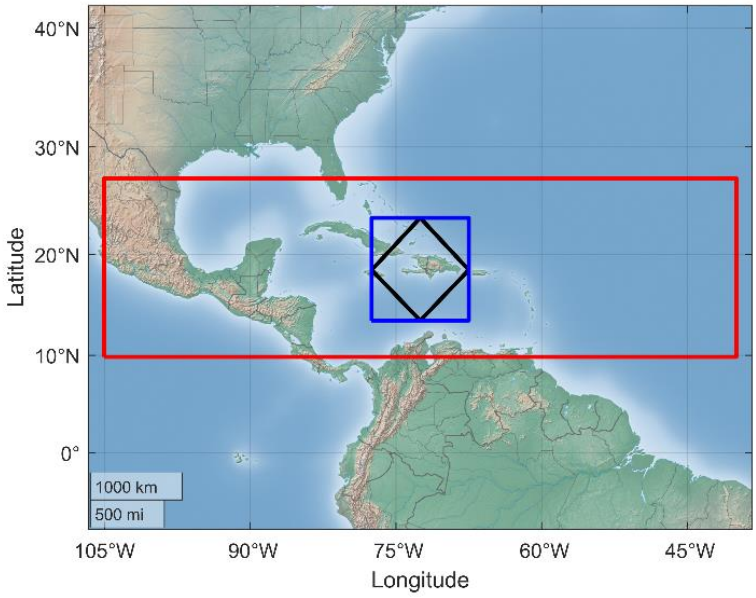


Figure 4.14 - Example of area of interest (in black) from which the minimum bounding rectangle (in blue) and the resulting safe area (in red) are defined.

The overflight analysis block outputs a subset of satellites passing over the AOI within the required time without the need of manoeuvring, and their corresponding time to overflight (t_{of}). The number of such satellites is indicated as N_{NM}, which can go from 0 to N_{sat} in the most general case. The remaining satellites (whose number, N_M, is equal to $N_{sat} - N_{NM}$) are input to the overflight manoeuvre design block which implements the ground-track adjustment method from [151] customized to account for the effective size of the optical and radar sensor swath-width. The operations conducted in this block, which are described in detail in the following subsection, output the required impulsive ΔV and the t_{of} for each of the N_M satellites.

Finally, such solution in terms of ΔV and t_{of} for the manoeuvring satellites, as well as the values of t_{of} for the non-manoeuvring satellites (for which ΔV is 0 km/s), are input to the optimization block which implements the multi-objective optimization strategy described in detail in the following subsection, to identify the most convenient solution to collect data from the AOI.

Ground-track adjustment methodology

This subsection describes the approximate analytical method derived by Zhang et al. [151] (see "*Fundamentals on the ground-track adjustment method*") and how it is customized to account for the coverage of remote sensing sensors installed on board Earth observation satellites with pyramidal-shape FOV (see "*Sensor-aware correction approach*").

Fundamentals on the ground-track adjustment method

Given a site of interest on ground (P_S), identified by its geodetic latitude (λ_S) and longitude (ϕ_s), and a satellite on an initial circular or elliptical orbit, the method allows defining a dual coplanar impulsive manoeuvre producing a transfer to a final circular orbit which passes over the site itself. The method's approximation stems from considering the Earth oblateness (by means of the J2 term of the gravitational potential) while discarding the perturbations due to drag, third bodies, solar radiation pressure, and higher order gravitational terms. The overflight condition is determined by the passage of the satellite's ground track on the Earth site.

In the case of a satellite on a circular orbit, a first solution must be obtained under the Keplerian motion assumption. The Keplerian time to overflight ($t_{of,k}$) can be obtained exploiting the Greenwich mean sidereal time (GMST) variation through Eq. (4.23),

$$t_{of,k} = \frac{(\alpha_{Gt} - \alpha_{G0}) + 2\pi\,(D-1)}{\omega_{\oplus}} \tag{4.23}$$

where $\omega_{\oplus} = 7.2921158553\,10^{-5}\,rad/s$ is the Earth rotation rate, D is still the maximum time to the expected overflight, α_{Gt} and α_{G0} are the GMST angles when the subsatellite point is P_S and at the initial time, respectively. The same quantity can also be computed using Eq. 4.24, which derives from the assumption of using an Hohmann transfer to reach the target orbit.

$$t_{of,k} = \pi\sqrt{\frac{\left(a_0 - a_{S,k}\right)^3}{8\mu}} + \left(u_t - u_{00} - \pi + 2\pi N_R\right)\sqrt{\frac{{a_{S,k}}^3}{\mu}} \tag{4.24}$$

The first term in Eq. (4.24) is the transfer time for the Hohmann-transfer arc (t_H), which depends on the semimajor axis of the initial orbit (a_0), the unknown semimajor axis of the final orbit under Keplerian motion assumption ($a_{S,k}$), and the Earth gravitational parameter (μ). With regards to the other terms, u_{00} is the argument of latitude of the satellite on the initial orbit at the initial time, u_t is the argument of latitude of the satellite on the final orbit when passing over P_S, and N_R is the number of orbit revolutions that minimize the required ΔV. In particular, u_t and N_R are computed according to Eq. (4.25),

$$\begin{aligned} u_t &= \sin^{-1}\left(\frac{\sin\varphi_s}{\sin i_0}\right) \\ N_R &= \frac{1}{2\pi}\left(t_{of,k}\sqrt{\frac{\mu}{a_0^3(1+e_0)^3}} - u_t + u_{00}\right) \end{aligned} \tag{4.25}$$

where i_0 is the inclination of the initial orbit and e_0 is the eccentricity of the initial orbit (which is equal to 0 or the case of initial circular orbit).

Given the equivalence between Eq. (4.23) and (4.24), and since $t_{of,k}$ can be computed from equation Eq. (4.23), $a_{S,k}$ can be derived solving equation (4.26).

$$\left[\frac{\pi}{4} + \left(u_t - u_{00} - \pi + 2\pi N_R\right)\right]\left(\sqrt{a_{S,k}}\right)^3 + \frac{3\pi}{4}a_0\sqrt{a_{S,k}} - t_{of,k}\sqrt{\mu} = 0 \tag{4.26}$$

The J_2 effect cannot be neglected when the time of flight is of the order of several days. Under the linear J_2 perturbation, the semi-major axis (a), the eccentricity (e), and inclination (i) remain constant, while the argument of perigee (ω), the right ascension of the ascending node (Ω), and the mean anomaly (M) change according to Eq. (4.27),

$$\dot{\omega}_{J_2} = C_{J_2}\left(2 - \frac{5}{2}\sin^2 i\right)/(1-e^2)^2$$

$$\dot{\Omega}_{J_2} = -C_{J_2}\cos i/(1-e^2)^2 \tag{4.27}$$

$$\dot{M}_{J_2} = C_{J_2}\left(1 - \frac{3}{2}\sin^2 i\right)/(1-e^2)^{3/2}$$

where $C_{J_2} = 1.5\, J_2\, R_\oplus^2\sqrt{\mu}\, a^{-7/2}$ and $J_2 = 1.082627 \cdot 10^{-3}$.

The value of t_{of} under the J_2 effect can thus be obtained using the GMST variation through Eq. (4.28), as well as by means of the approximate expression (4.29).

$$t_{of} = \frac{\alpha_{Gt} - \alpha_{G0} + 2\pi(D-1)}{\omega_\oplus - \dot{\Omega}_{J_2}} \tag{4.28}$$

$$t_{of} \approx \chi_{H_s}\pi\sqrt{\frac{(a_0+\bar{a})^3}{8\mu}} + \chi_s(u_t - u_{00} - \pi + 2\pi N_R)\sqrt{\frac{\bar{a}^3}{\mu}} \tag{4.29}$$

In Eq. (4.29), $\bar{a}$ is the unknown mean semimajor axis of the final orbit while χ_{H_s} and χ_s are correction factors which can be computed through Eq. (4.30).

$$\chi \triangleq \left[1 + \left(\dot{M}_{J_2} + \dot{\omega}_{J_2}\right)\sqrt{\bar{a}^3/\mu}\right]^{-1} \tag{4.30}$$

Specifically, χ_s and χ_{H_s} are obtained by replacing a and e in Eq. (4.27) with $a_{s,k}$ and $e = 0$ and with $a_H = \left(a_{s,k} + a_0\right)/2$ and $e_H = \left|a_{s,k} - a_0\right|/\left(a_{s,k} - a_0\right)$, respectively.

The equivalence between Eq. (4.28) and (4.29) leads to a cubic equation in terms of $\sqrt{\bar{a}}$, which can be used to obtain the mean semi-major axis of the final orbit under the J_2 effect:

$$\left[\frac{\pi}{4}\,\chi_{H_s} + \chi_s(u_t - u_{00} - \pi + 2\pi N_R)\right]\left(\sqrt{\bar{a}}\right)^3 + \frac{3\pi}{4}\chi_{H_s} a_0\sqrt{\bar{a}} - t_{GJ_2}\sqrt{\mu} = 0 \tag{4.31}$$

The other orbital elements of the final orbit are identical to those of the initial orbit: they are first set as mean orbital elements and then transformed into osculating ones. The osculating semi-major axis of the final orbit (a_f) is finally used to determine the ΔV_{tot} required by the manoeuvring satellites considering a two-impulse Hohmann-transfer manoeuvre.

$$\Delta V_{tot} = \sqrt{\frac{\mu}{a_0}}\left|\sqrt{\frac{2a_f}{a_0 + a_f}} - 1\right| + \sqrt{\frac{\mu}{a_f}}\left|1 - \sqrt{\frac{2a_0}{a_0 + a_f}}\right| \tag{4.32}$$

The first impulse is applied at the initial time while the second impulse is applied at the apogee of the transfer orbit, whose time instant is given by the first term at the right-hand side of Eq. (4.29).

When starting from an initial elliptical orbit, $t_{of,k}$ is computed using Eq. (4.33).

$$t_{of,k} = (\pi - M_{00})\sqrt{\frac{a_0^3}{\mu}} + \pi\sqrt{\frac{\left[a_0(1 + e_0) + a_{s,k}\right]^3}{8\mu}} + (u_t - \omega_0 + 2\pi N_R)\sqrt{\frac{{a_{s,k}}^3}{\mu}} \tag{4.33}$$

where e_0 is the eccentricity of the initial orbit, M_{00} is the mean anomaly of the initial orbit at the initial time, and ω_0 is the argument of perigee of the initial orbit. In Eq. (4.33), the first term stands for the coasting time $t_{coasting}$ required to reach the apogee of the initial orbit and the second term stands for the Hohmann transfer time t_H. As in the previous case, the Keplerian semi-major axis of the final orbit, $a_{s,k}$, is first obtained solving Eq. (4.34), which derives from the equivalence between Eq. (4.33) and Eq. (4.23).

$$\left[\frac{\pi}{4} + (u_t - \omega_0 + 2\pi N_r)\right]\left(\sqrt{a_{s,k}}\right)^3 + \frac{3\pi}{4}a_0(1+e_0)\sqrt{a_{s,k}} - \left[t_{of,k}\sqrt{\mu} - (\pi - M_{00})\right]\sqrt{a_0^3} = 0 \tag{4.34}$$

The value of $a_{s,k}$ is used to determine $\dot{\omega}_{J_2}$, $\dot{\Omega}_{J_2}$ and $\dot{M}_{J_2}$ through Eq. (4.26) while t_{of} is calculated through Eq. (4.27). As in the case of initial circular orbit, a value of t_{of} which accounts for the J_2 effect can be obtained using Eq. (4.35).

$$t_{of} \approx t_{coasting} + \chi_{H_s}\pi\sqrt{\frac{[a_0(1+e_0)+\bar{a}]^3}{8\mu}} + \chi_s(u_t - \omega_0 + 2\pi N_R)\sqrt{\frac{\bar{a}^3}{\mu}} \tag{4.35}$$

In Eq. (4.35) $t_{coasting}$ is obtained using numerical propagation and selecting Runge-Kutta 89 as integrator, while the parameters χ_s and χ_{H_s} are given by Eq. (4.30). Specifically, χ_s is obtained using the Keplerian solution $a_{s,k}$ in Eq. (4.27) and χ_{H_s} is computed by replacing a and e in Eq. (4.27) with $a_{H_s} = \left[a_{s,k} + a_0(1+e_0)\right]/2$ and $e_{H_s} = \left[a_{s,k} - a_0(1+e_0)\right]/\left[a_{s,k} + a_0(1+e_0)\right]$.

Given the equivalence between Eq. (4.35) and Eq. (4.28), a cubic equation in terms of $\sqrt{\bar{a}}$ can be again formulated, whose solution allows computing the desired mean semi-major axis of the final orbit.

$$\left[\frac{\pi}{4}\chi_{H_s} + \chi_s(u_t - \omega_0 + 2\pi N_R)\right]\left(\sqrt{\bar{a}}\right)^3 + \frac{3\pi}{4}\chi_{H_s}a_0(1+e_0)\sqrt{\bar{a}} - \left(t_{of} - t_{coasting}\right)\sqrt{\mu} = 0 \tag{4.36}$$

The other orbital elements of the final orbit are identical to those of the initial orbit at $t_{coasting}$. These orbital parameters are set as mean orbital elements and then they are transformed into osculating ones. Again, the osculating semimajor axis, a_f, is used to determine the value of ΔV_{tot} considering a Hohmann transfer manoeuvre and applying the first impulse at the apogee, through Eq. (4.37).

$$\Delta V_{tot} = \sqrt{\frac{\mu}{a_0(1+\ e_0)}}\left|\sqrt{\frac{2a_f}{a_0(1+\ e_0)+a_f}}-\sqrt{1-e_0}\right| + \sqrt{\frac{\mu}{a_f}}\left|1-\sqrt{\frac{2a_0(1+e_0)}{a_0(1+\ e_0)+a_f}}\right| \quad (4.37)$$

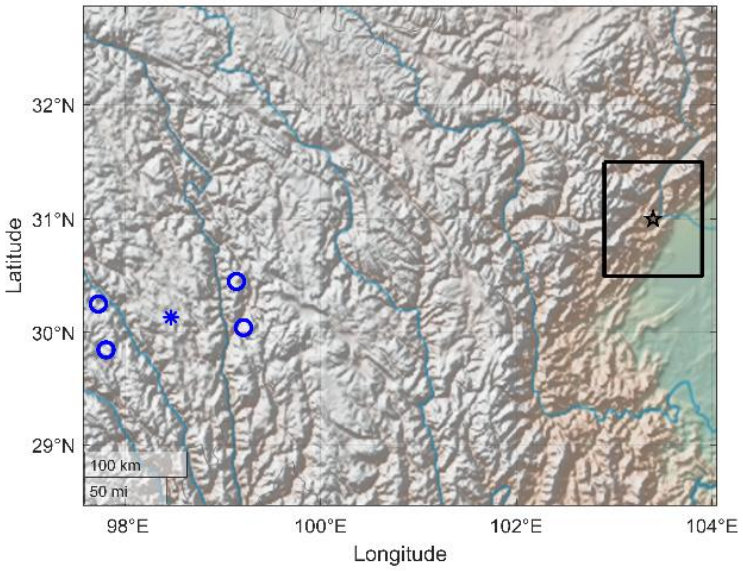
The first impulse is applied at the apogee of the initial orbit, i.e., at $t_1 = t_{coasting}$, while the second impulse occurs at the apogee of the transfer orbit, i.e., at $t_2 = t_{coasting} + t_H$.

Sensor-aware correction approach

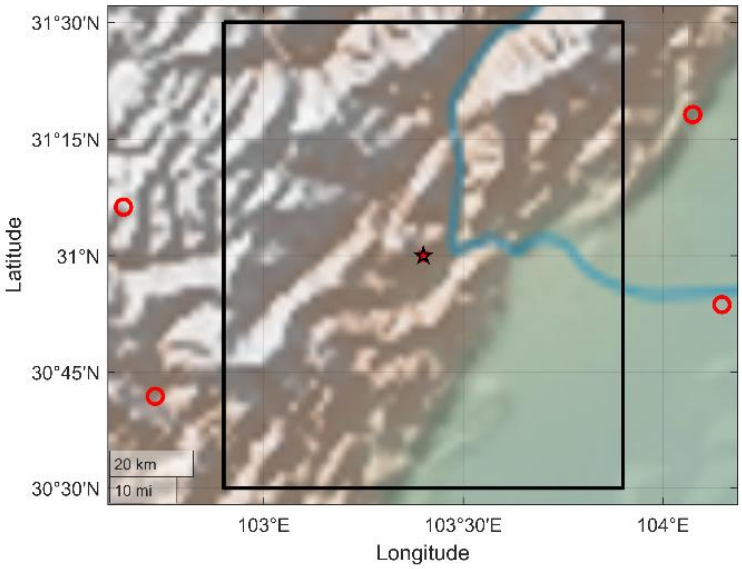
The analytical method, presented by Zhang et al. in [151] and briefly recalled in the previous subsection, leads to an approximate solution and a residual distance between the target and the subsatellite point of few kilometres. Such a distance might not be representative of the actual observability of the AOI from a sensor-equipped satellite as the target might not be within the sensor footprint. For this reason, Zhang et al. [151] extended the analytical solutions even when the satellite is equipped with a nadir-pointing conical sensor. To accomplish that, they applied the analytical method so that the target is located at the boundary of the conical sensor footprint, resulting in a reduction in fuel consumption. In this PhD thesis, the two-impulse method derived by Zhang et al. [151] is modified assuming that the satellite is equipped with either a nadir pointing optical sensor or an off-nadir pointing radar, both with a pyramidal-shaped FOV. To be conservative, the analytical method is applied so that the target is placed at the centre of the optical sensor or radar footprint.

The following procedure is applied to all the N_M satellites, as identified by the overflight analysis block of the proposed architecture (see Figure 4.12). For these satellites, at each time instant of their orbit ephemerides, there is no intersection between the AOI, whose centre coordinates are indicated as $P_{AOI}(\lambda_{AOI}, \phi_{AOI})$, and the footprint whose centre coordinates are indicated as $P_{FP}(\lambda_{FP}, \phi_{FP})$. Examples are shown in Figure 4.15 (a) and in Figure 4.16 (a).

For the case of an optically-equipped satellite, given the nadir pointing assumption, the subsatellite point $P_{SSP}(\lambda_{SSP}, \phi_{SSP})$ coincides with P_{FP}. Thus, the ground track adjustment problem described in the previous subsection is solved by setting $P_{AOI} \equiv P_{SSP} \equiv P_{FP}$ as the Earth site. This condition should lead to an overflight as shown in Figure 4.15 (b), aside from residual errors due to the inherent approximations of the method in [151].



(a)



(b)

Figure 4.15 - Example of unsuccessful overflight by the satellite's optical sensor (a) and successful overflight by the satellite's optical sensor (b). The footprint of the optical sensor is shown in blue: the empty dots are the vertices of the footprint and the asterisk is the sub-satellite point. Black lines mark the area of interest, and the black star represents the barycentre of the area of interest. The shifted footprint of the optical sensor is depicted in red.

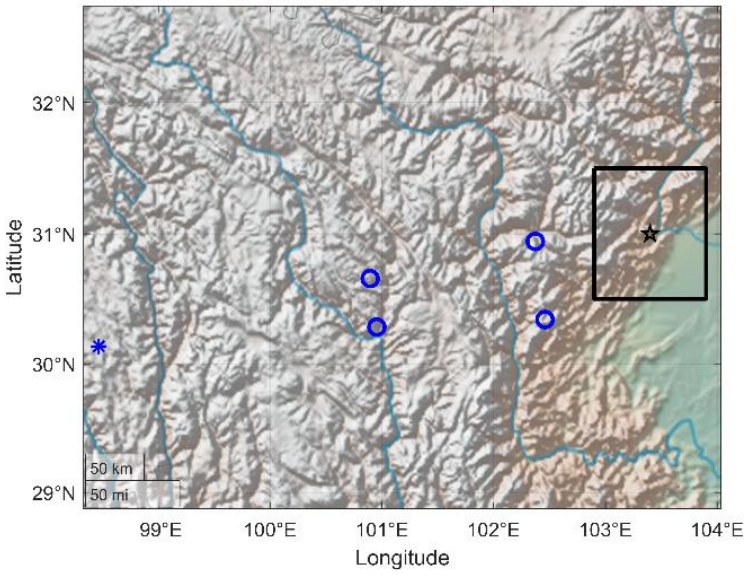
In the case of a radar-equipped satellite, given the off-nadir pointing assumption, the ground track adjustment problem described in the previous subsection must be solved by setting as Earth site the P_{SSP} at which $P_{FP} \equiv P_{AOI}$. To this aim, the ground distance ($d_{SPP,FP}$) between P_{SSP} and P_{FP} on the surface of the Earth modelled as an ellipsoid, whose semi-major axis is 6378.137 km and whose eccentricity is 0.0818, is calculated according to [159]. It is also required to compute the off-nadir angle of P_{FP} with respect to P_{SSP} (ϑ_{FP}) using the Eq. (4.38). Please note that the inverse tangent function in (4.38) is the 2-argument arctangent.

$$\theta_{FP} = tan^{-1}\left(\frac{\sin(\lambda_{SSP} - \lambda_{FP})\cos(\phi_{SSP})}{\cos(\phi_{FP})\sin(\phi_{SSP}) - \sin(\phi_{FP})\cos(\phi_{SSP})\cos(\lambda_{SSP} - \lambda_{FP})}\right) \quad (4.38)$$

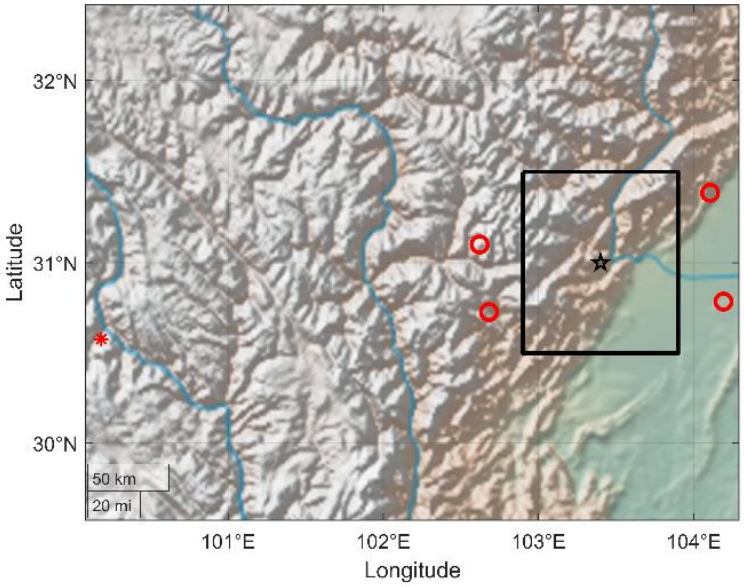
Finally, the geodetic coordinates of the desired P_{SSP} can be computed using Eq. (4.39),

$$\phi_{SSP} = \phi_{AOI} + \frac{d_{SSP,FP}}{R}\cos(\theta_{FP})$$
$$\lambda_{SSP} = \lambda_{AOI} + \frac{d_{SSP,FP}}{R}\sin(\theta_{FP})\frac{1}{\cos(\phi_{FP})} \quad (4.39)$$

where R is the Earth radius. These equations should lead to an overflight as shown in Figure 4.16 (b), aside from residual errors due to the inherent approximations of the method in [151].



(a)



(b)

Figure 4.16 - Example of unsuccessful overflight by the satellite's radar (a) and successful overflight by the satellite's radar (b). The footprint of the radar is shown in blue: the empty dots are the vertices of the footprint and the asterisk is the sub-satellite point. Black lines mark the area of interest, and the black star represents the barycentre of the area of interest. The shifted footprint of the radar is depicted in red.

For satellites equipped with an optical sensor, illumination conditions are taken into account and, thus, only overflight solutions that occur during the daytime are considered. To discriminate between day and night, dawn and dusk are computed using standard astronomical algorithms [160] and considering civil twilight (i.e., the Sun at 6° below the horizon) as the criterion for the start and end of the daylight period.

Multi-objective optimal ground-track adjustment

Multi-objective optimization is applied in many real-world engineering problems where objectives under consideration conflict with each other, and optimizing a particular solution with respect to a single objective can result in unacceptable results with respect to the other objectives [161], [162], [163]. Multi-objective optimization problems provide a set of equally optimal solutions, referred to as Pareto-optimal solutions or non-dominated solutions, which form the Pareto optimal front, as

conceptually shown in Figure 4.17. A solution is said to be non-dominated if it cannot be improved with respect to any objective without worsening at least one other objective. In contrast, dominated solutions are those for which at least one non-dominated solution that maximises/minimizes all the objectives exists.

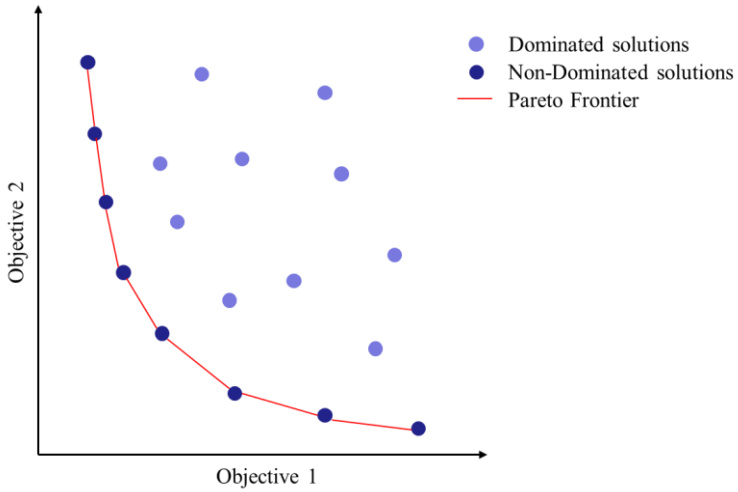


Figure 4.17 - Conceptual representation of dominated solutions, non-dominated solutions and Pareto Frontier for a two-objective optimization problem.

After obtaining the set of non-dominated solutions, a decision maker can use higher-level qualitative considerations or implement an automatic decision-making method to make a choice [164].

In this work, a set of N_{sat} solutions is obtained to cover the desired AOI within the maximum assigned time frame, D. Each solution is identified by t_{of} (which is computed using the overflight analysis block for non-manoeuvring satellites and the overflight manoeuvre design block for manoeuvring satellites) and ΔV (which is 0 for non-manoeuvring satellites while it is computed within the overflight manoeuvre design block for manoeuvring satellites).

To find a subset of non-dominated solutions to this problem and then to select a single best solution, a multi-attribute decision-making (MADM) method [165] is applied in the Optimization block of Figure 4.12. In general, considering a set of

m alternatives, $a_i(i = 1,\ 2, \dots ,\ m)$, evaluated with respect to a set of n attributes or criteria, $C_j(j = 1,\ 2, \dots ,\ n)$, a decision matrix for m alternatives and n attributes can be defined as in [165]:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots_1 & x_{1n} \\ x_{21} & x_{22} & \cdots_1 & x_{2n} \\ \cdots_1^1 & \cdots_1^1 & \cdots_1 & \cdots_1^1 \\ x_{m1} & x_{m2} & \cdots^1 & x_{mn} \end{bmatrix} \tag{4.40}$$

where x_{ij} stands for the performance value of the $i-th$ alternative in terms of the $j-th$ criterion. A weighting vector representing the relative importance of the attributes can also be set as:

$$W = (w_1 \quad w_2 \quad \cdots \quad w_n) \tag{4.41}$$

One of the most widely used MADM technique is the simple additive weighting (SAW) method, which consists in performing a weighted sum of the performance values of each alternative over all attributes. The SAW method typically requires normalizing the decision matrix X to allow a comparable scale for all its values, as follows.

$$\begin{cases} \bar{r}_{ij} = \dfrac{x_{ij}}{\max\limits_i x_{ij}} & \text{if} \max\limits_i x_{ij} \text{ is preferable} \\ \bar{r}_{ij} = \dfrac{\min\limits_i x_{ij}}{x_{ij}} & \text{if } \min\limits_i x_{ij} \text{ is preferable} \end{cases} \tag{4.42}$$

Therefore, $\bar{r}_{ij}$ $(0 \ \leq \bar{r}_{ij} \leq 1)$ is defined as the normalized performance value of the $i-th$ alternative when it is evaluated in terms of the $j-th$ criterion. The preference score of each alternative (A_i) can be calculated by Eq. 4.43 as in [165]:

$$A_i = \sum_{j=1}^{n} w_j \bar{r}_{ij} \ \ for\ i = 1,2, \dots, m \tag{4.43}$$

The higher the preference score A_i, the more preferable the alternative a_i is. In the case under consideration, the X matrix will have [m x 2] dimension where m is equal to N_{Sat}.

$$X = \begin{bmatrix} \Delta V_1 & t_{of_1} \\ \Delta V_2 & t_{of_2} \\ \ldots_1^1 & \ldots_1^1 \\ \Delta V_m & t_{ofm} \end{bmatrix} \tag{4.44}$$

Since both the ΔV and the t_{of} are to be minimised, the minimum value of the first and second column are considered, ΔV_{min} and $t_{of,min}$, to define the normalized matrix, as follows,

$$\bar{r} = \begin{bmatrix} \overline{\Delta V_1} & \overline{t_{of1}} \\ \overline{\Delta V_2} & \overline{t_{of2}} \\ \ldots & \ldots \\ \overline{\Delta V_m} & \overline{t_{ofm}} \end{bmatrix} \tag{4.45}$$

where: $\overline{\Delta V_i} = \Delta \mathrm{V}_{min}/\Delta V_i$ and $\overline{t_{of,i}} = t_{of,min}/t_{of,i}$. It is worth highlighting that when non-manoeuvring overflights lie on the Pareto frontier, the simple additive weighting (SAW) method is applied considering a ΔV_{min} equal to 10^{-6} km/s to avoid division by 0 in the normalization. Once the ΔV and the t_{of} have been normalized, a pair of weights (w_1, w_2) is chosen for the ΔV and the t_{of} so that the score for each of the m alternatives can be computed as follows.

$$a_i = \overline{\Delta V_i} w_1 + \overline{t_{of,i}} w_2 \tag{4.46}$$

Regarding the weights, three options are considered for the numerical performance assessment of the proposed methodology presented in Section 4.2.3. First, the ΔV and the t_{of} are considered to be equally important, namely $(w_1, w_2) = (0.5, 0.5)$. Second, an emergency scenario is assumed giving more importance to the t_{of}, thus having $(w_1, w_2) = (0.4, 0.6)$. Third, a stricter requirements is posed on fuel consumption, thus setting $(w_1, w_2) = (0.6, 0.4)$. The choice of these limit values of the weights is justified by the absence of change in the optimal solution when considering values larger than 0.6 or smaller than 0.4 noticed from all the conducted numerical tests.

4.2.3 Numerical results

Three analyses are carried out to validate and assess the proposed methodology. The first analysis considerers a set of five simulated LEO satellites, three equipped with an optical sensor and two with a radar; it aims to assess the effect on methodology performance of choosing different AOIs, centred at a mid-latitude, at a high latitude, and at a low latitude respectively. The second analysis considers the same set of five satellites and an AOI centred at a mid-latitude; in this case, the performance of the proposed methodology has been evaluated considering three different payload configurations: three satellites equipped with optical sensors and two with radars, all satellites equipped with optical sensors and all satellites equipped with radar. Finally, the third analysis applies the proposed methodology to a real-world scenario, considering a set of five disaster monitoring satellites. The three analyses are discussed in the following dedicated subsections.

Effect of AOI position

A set of five simulated satellites is defined with the following physical features: m = 850 kg, A_{drag} = 3.86 m^2 and C_d = 2.2: this results in a ballistic coefficient (*BC*) equal to 0.01 m^2/kg (this is in fact a common value for Earth observation operative satellites [166]). The initial orbital parameters of the simulated satellites are reported in Table 4.3, assuming they fly on circular orbits (i.e., e = 0): the semimajor axis and inclination values have been chosen considering typical orbital characteristics of Earth observation satellites in LEO; the right ascension of the ascending node values have been chosen to equally cover the whole interval of [0, 360] °; the argument of perigee and the initial true anomaly have been set equal to 0°. Without losing generality, the orbital parameters of all the satellites are assumed to refer to the same epoch, i.e., July 1, 2019 at 08:00:00.

Table 4.3 - Initial orbital elements of the simulated LEO satellites.

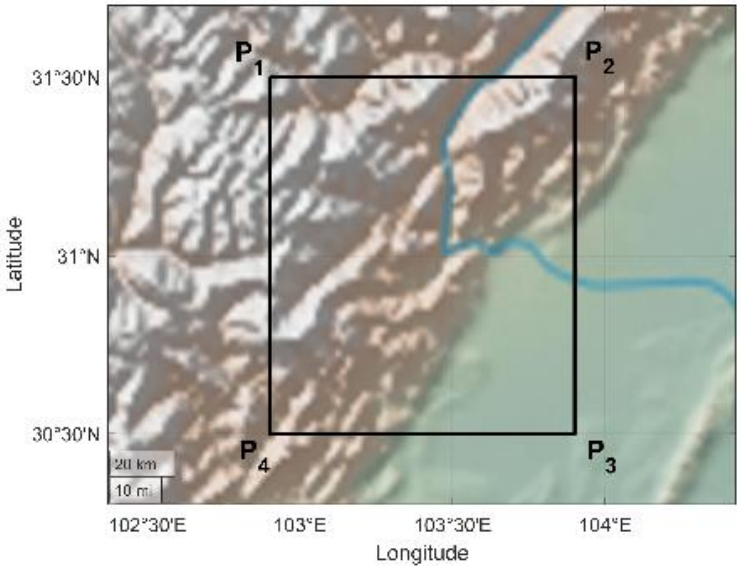
Satellite #	$\boldsymbol{a}$ $[\boldsymbol{km}]$	$\boldsymbol{e}$ $[-]$	$\boldsymbol{i}$ [°]	$\boldsymbol{\Omega}$ [°]	$\boldsymbol{\omega}$ [°]	$\boldsymbol{\nu}$ [°]
1	6828.14	0	97.20	0	0	0
2	6928.14	0	92.00	72	0	0
3	7028.14	0	97.99	144	0	0
4	7128.14	0	71.00	216	0	0
5	7228.14	0	98.82	288	0	0

The first, second, and fifth satellites are assumed to be equipped with an optical sensor with α = 12° and β = 4°. The third and fourth satellites instead have an on-board radar with γ = 10°, θ_{min} = 20° and θ_{max} = 30°.

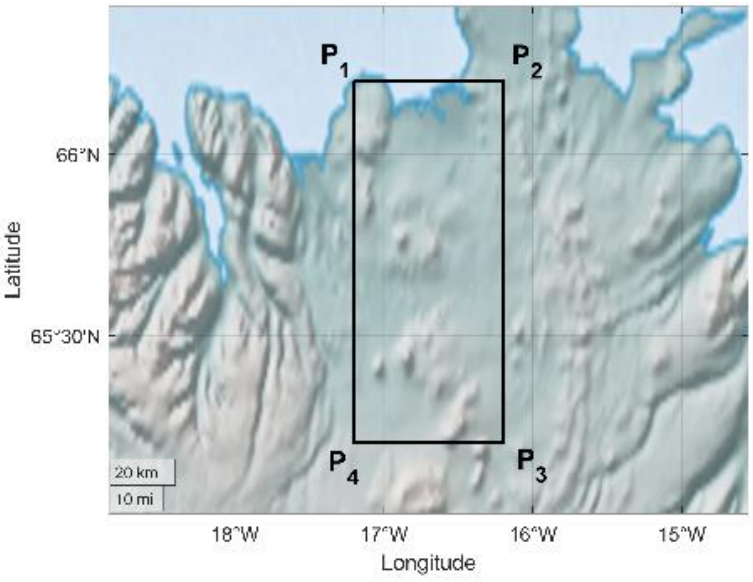
The selected AOIs centred at a mid-latitude, a high latitude, and a low latitude are depicted in Figure 4.18. Without losing generality, square-shaped AOIs are considered, characterized by a width of 1° in both latitude and longitude: specifically, the geodetic coordinates of the AOI vertices (P_1, P_2, P_3 and P_4, in clock-wise order starting from the upper-left point) are reported in Table 4.4. The architecture described in Section 4.2.2 is applied setting D to 2 days.

Table 4.4 - Geodetic coordinates of AOI vertices at middle, high, and low latitudes

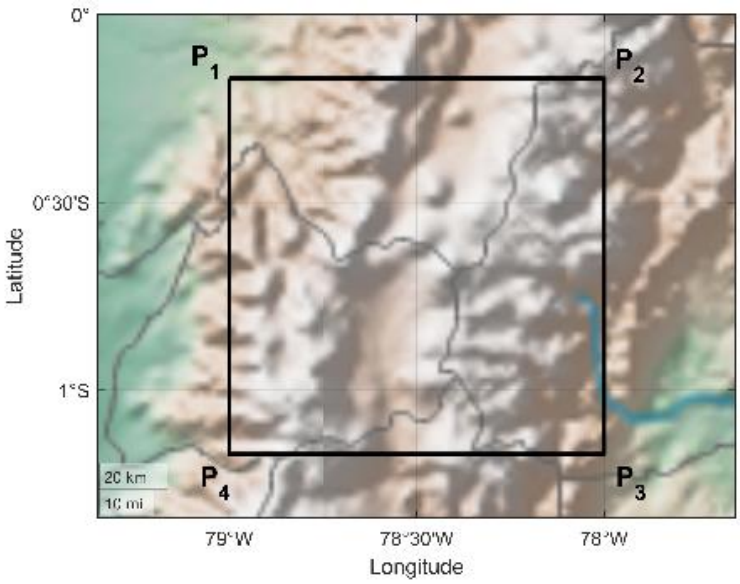
	$\boldsymbol{P_1(\lambda_1, \phi_1)}$	$\boldsymbol{P_2(\lambda_2, \phi_2)}$	$\boldsymbol{P_3(\lambda_3, \phi_3)}$	$\boldsymbol{P_4(\lambda_4, \phi_4)}$
Mid-latitudes AOI	(102.9°, 31.5°)	(103.9°, 31.5°)	(103.9°, 30.5°)	(102.9°, 30.5°)
High latitudes AOI	(-17.2°, 66.2°)	(-16.2°, 66.2°)	(-16.2°, 65.2°)	(-17.2°, 65.2°)
Low latitudes AOI	(-79°, -0.17°)	(-78°, -0.17°)	(-78°, -1.17°)	(-79°, -1.17°)


P1
P2
P3
P4
31°30'N
31°N
30°30'N
Latitude
20 km
10 mi
102°30'E
103°E
103°30'E
104°E
Longitude

(a)


P1
P2
P3
P4
66°N
65°30'N
Latitude
20 km
10 mi
18°W
17°W
16°W
15°W
Longitude

(b)



(c)

Figure 4.18 - AOIs at middle, high, and low latitudes.

With regards to the AOI centred at mid-latitude, the results of the proposed architecture are reported in Table 4.5. For the first, second and fifth satellites, which are equipped with an optical sensor, illumination conditions have been taken into account. In particular, for the second and fifth satellites, only descending passes have been considered since ascending passes happen at night. For the first satellite, instead, all solutions have been taken into account, since both ascending and descending passes occur during the daytime.

Table 4.5 - 4.6 ΔV_{tot} and t_{of} for the AOI centred at a mid-latitude, considering $D = 2$. Only ascending (A) and descending (D) passages compatible with illumination constraints for optical sensors are reported.

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]
1	1 (D)	0.890	2.8
	1 (A)	0	14.2
	2 (D)	0.0330	24 + 2.8
	2 (A)	0.0588	24 + 14.2
2	1 (D)	0	19.3
	2 (D)	0	24 + 19.1

3	1 (D)	0.300	12.2
	1 (A)	0.0590	24.0
	2 (D)	0.0187	24 + 12.2
	2 (A)	0.0508	24 + 24.0
4	1 (A)	0.393	5.9
	1 (D)	0	15.6
	2 (A)	0.0323	24 + 5.7
	2 (D)	0.0682	24 + 15.5
5	1 (D)	0.115	21.8
	2 (D)	0.0419	24 + 21.8

The score evaluation for the subset of non-dominated solutions, which form the Pareto optimal front depicted in Figure 4.19, is reported in

Table 4.7 considering the three selected sets of weights. By assigning equal weights to ΔV_{tot} and t_{of}, the optimal solution, i.e., the one characterized by a higher score, corresponds to the observation occurring in the ascending phase of the first overflight window of the first satellite without the need to manoeuvre and a time to overflight of 14.2 h. Increasing w_1 to 0.6, the non-manoeuvring solution is still the best one. Increasing w_2 to 0.6, the best solution is the one foreseeing a manoeuvre of the first satellite, considering the descending pass of the first overflight window, with a ΔV_{tot} equal to 0.8896 km/s and a t_{of} equal to 2.8 h.

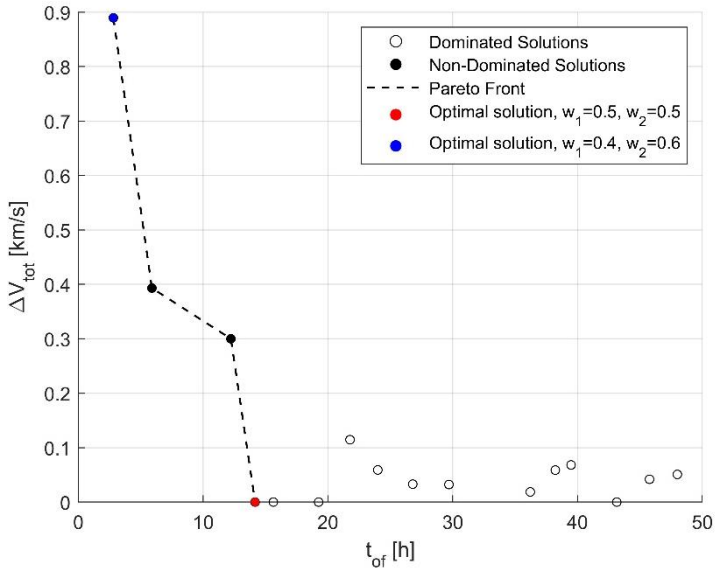


Figure 4.19 - Non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the SAW method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) for the area of interest at mid-latitudes. Same solution to the former for (w_1, w_2) equal to (0.6, 0.4).

Table 4.7 Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
1	1 (D)	0.890	2.8	0.600	0.500	0.400
	1 (A)	0	14.2	0.519	0.599	0.679
3	1 (D)	0.300	12.2	0.138	0.115	0.0918
4	1 (A)	0.393	5.9	0.286	0.238	0.191

Results for the case of an AOI centred at a high latitude are synthetized in Table 4.8. For the first, second and fifth satellites, which are equipped with an optical sensor, illumination conditions are taken into account. In this case, the ascending and descending passes of the aforementioned satellites take place during the daytime. Therefore, there are no discarded solutions.

Table 4.8 - ΔV_{tot} and t_{of} for the area of interest at high latitudes, considering $D = 2$. Only passages (ascending and/or descending) compatible with illumination conditions are reported.

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]
1	1 (D)	0.295	11.6
	1 (A)	0.0806	21.4
	2 (D)	0.0528	24 + 11.6
	2 (A)	0	24 + 21.5
2	1 (A)	0.421	3.0
	1 (D)	0.140	15.6
	2 (A)	0.0385	24 + 3.0
	2 (D)	0.0512	24 + 15.5
3	1 (A)	0.365	7.4
	1 (D)	0.0987	20.9
	2 (A)	0.118	24 + 7.4
	2 (D)	0.0686	24 + 20.9
4	1 (A)	0	17.0
	1 (D)	0	20.4
	2 (A)	0.0665	24 + 16.8
	2 (D)	0.0630	24 + 20.2
5	1 (D)	0.0951	7.1

	1 (A)	0.138	16.4
	2 (D)	0	24 + 7.1
	2 (A)	0.0415	24 + 16.4

Figure 4.20 shows the results of the two-objective optimization (i.e., the dominated and non-dominated solutions and the Pareto frontier) as well as the optimal solution obtained by applying the SAW method for pairs of weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6). As can be seen in Figure 4.20, there is a non-manoeuvring overflight that lies on the Pareto frontier. Table 4.9 reports the score evaluation for the subset of non-dominated solutions, considering the three selected sets of weights. By assigning equal weights to ΔV_{tot} and t_{of}, the optimal solution corresponds to the observation occurring in the ascending pass of the first overflight window of the fourth satellite, without the need to manoeuvre. For the pair of weights (0.6, 0.4) the non-manoeuvring solution is still the best one. Increasing the weight assigned to t_{of} to $w_2 = 0.6$, it is advantageous to manoeuvre the second satellite, considering the ascending pass of the first overflight window.

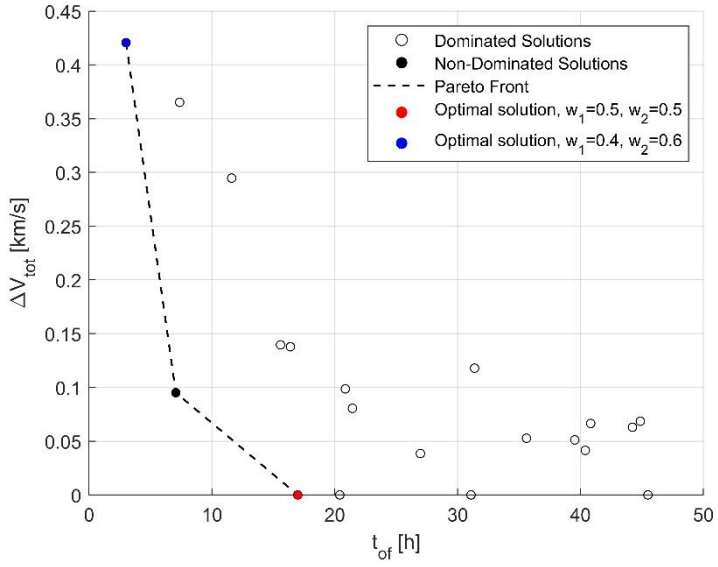


Figure 4.20 - Non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the SAW method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) for the area of interest at high latitudes. Same solution to the former for (w_1, w_2) equal to (0.6, 0.4).

Table 4.9 - Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
2	1 (A)	0.421	3.0	0.600	0.500	0.400
4	1 (A)	0	17.0	0.507	0.589	0.671
5	1 (D)	0.0951	7.1	0.256	0.214	0.171

Table 4.10 provides the results for the case of an AOI centred at a low latitude. For the first, second and fifth satellites, which are equipped with optical sensors, only ascending or descending passes are considered depending on the illumination conditions. It can be noticed that there are no non-manoeuvring overflights in this case.

Table 4.10 ΔV_{tot} and t_{of} for the area of interest at low latitudes, considering $D = 2$. Only passages (ascending and/or descending) compatible with illumination conditions are reported.

Satellite #	Day(A/D)	ΔV_{tot}(km/s)	t_{of}(h)
1	1 (A)	0.0328	14.6
	2 (A)	0.0735	24 + 14.6
2	1 (A)	0.191	7.4
	2 (A)	0.0397	24 + 7.4
3	1 (A)	0.123	12.4
	1 (D)	0.123	24 + 0.1
	2 (A)	0.0696	24 + 12.4
	1 (D)	0.556	4.8
4	1 (A)	0.178	17.1
	2 (D)	0.0454	24 + 4.6
	2 (A)	0.0422	24 + 16.9
5	1 (D)	0.303	9.8
	2 (D)	0.0713	24 + 9.8

Table 4.11 provides the score evaluation for the subset of non-dominated solutions, which form the Pareto optimal front depicted in Figure 4.21, considering the three selected sets of weights. The optimal solution obtained by assigning w_1 = 0.5 and w_2

= 0.5 corresponds to the observation occurring in the ascending pass of the first overflight window by the first satellite, with ΔV_{tot}= 0.0328 km/s and time to overflight = 14.6 h. The same solution is obtained for (w_1, w_2) equal to (0.6, 0.4). When assigning weights $w_1 = 0.4$ and $w_2 = 0.6$, it is preferable to manoeuvre the fourth satellite considering the descending pass of the first overflight window, with ΔV_{tot} = 0.556 km/s and t_{of} = 4.8 h.

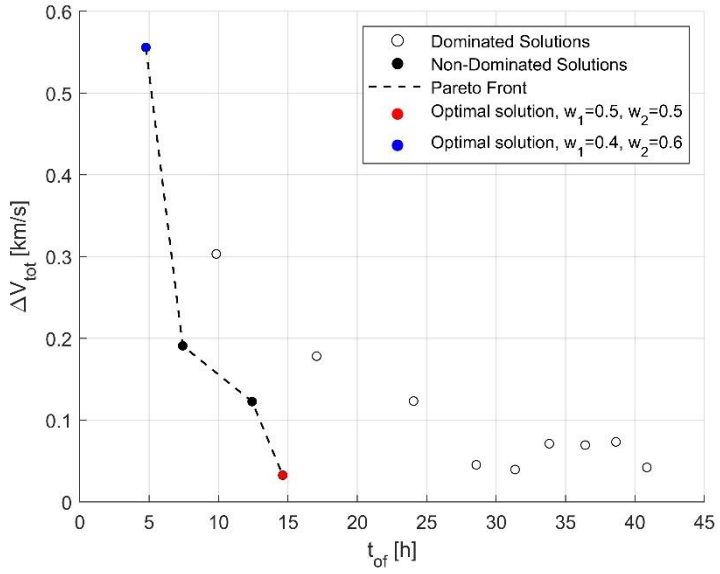


Figure 4.21 - Non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the SAW method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) for the area of interest at low latitudes. Same solution to the former for (w_1, w_2) equal to (0.6, 0.4).

Table 4.11 - Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
1	1 (A)	0.0328	14.6	0.596	0.663	0.730
2	1 (A)	0.191	7.4	0.454	0.407	0.360
3	1 (A)	0.123	12.4	0.337	0.326	0.314
4	1 (D)	0.556	4.8	0.624	0.530	0.435

To sum up, the following considerations can be made. The selection of an area of interest at higher latitudes increases the number of satellites that overfly the area of

interest without the need of manoeuvring: therefore, the number of satellites requiring manoeuvring will decrease, but the manoeuvred subset does not show any appreciable variations in ΔV_{tot} and t_{of}. Both at high and mid latitudes, the manoeuvring solution is favoured only if a stricter requirement is posed on responsiveness (i.e., w_2 = 0.6). In the present case, the number of solutions given as input to the optimization algorithm is greater for the area of interest at high latitudes. Indeed, for satellites equipped with an optical sensor, the number of solutions under consideration increases as latitude increases, since the selected initial epoch is July 1, 2019.

Effect of different payload configurations

The second analysis is conducted considering the same set of simulated satellites as those in the previous subsection. Three different payload configurations have been considered: three satellites equipped with an optical sensor and two with a radar (case A), all satellites equipped with an optical sensor (case B) and all satellites equipped with a radar (case C). The same specifications as in the previous subsection are set for the optical and radar sensors.

The chosen AOI, whose vertices have the following geodetic coordinates: P_1 (λ = 13.7°, ϕ = 41.15°), P_2 (λ = 14.7°, ϕ = 41.15°), P_3 (λ = 14.7°, ϕ = 40.55°) and P_4 (λ = 13.7°, ϕ = 40.55°), is depicted in Figure 4.22. D is again set equal to 2 days.

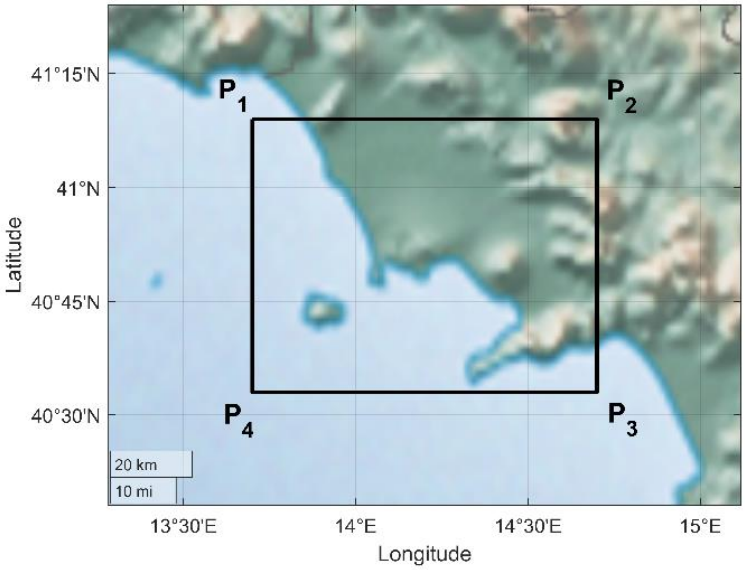


Figure 4.22 - AOI chosen for the second test case.

Results for case A are reported in Table 4.12. Specifically, it is assumed that the first, second, and third satellites have an optical sensor, and the third and fourth satellites have a radar. For optical sensor-equipped satellites, some solutions have been discarded depending on the illumination conditions. In particular, only descending passes have been considered for the second and the fifth satellites. For the first satellite, instead, all the observations happen during the daytime and, thus, both ascending and descending passes have been taken into account.

Table 4.12 - ΔV_{tot} and t_{of} for case A, considering $D = 2$. Only passages (ascending and/or descending) compatible with illumination conditions are reported.

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]
1	1 (D)	0.308	8.9
	1 (A)	0.0517	20.0
	2 (D)	0.0355	24 + 8.9
	2 (A)	0.0769	24 + 20.0
2	1 (A)	0.822	4.9
	2 (A)	0.163	24 + 4.6
3	1 (A)	0.384	5.8
	1 (D)	0.0349	18.3
	2 (A)	0.108	24 + 5.8
	2 (D)	0.0394	24 + 18.3

4	1 (A)	0.272	12.2
	1 (D)	0.122	21.2
	2 (A)	0.0558	24 + 12.0
	2 (D)	0.0278	24 + 21.0
5	1 (D)	0.903	4.2
	2 (D)	0.117	24 + 4.2

Table 4.13 provides the score evaluation for the subset of non-dominated solutions, which form the Pareto optimal front shown in Figure 4.23, considering the three selected sets of weights. The best solution obtained by applying the SAW method with $w_1 = 0.5$ and $w_2 = 0.5$ corresponds to the observation occurring in the descending pass of the second overflight window by the fourth satellite, with a time to overflight of 45 h and ΔV_{tot} = 0.0278 km/s. This solution is obviously confirmed as the best for the pair of weights pair (0.6, 0.4). Instead, when assigning more weight to the overflight time ($w_2 = 0.6$), it is advantageous to manoeuvre the fifth satellite considering the descending pass of the first overflight window with a ΔV_{tot} = 0.903 km/s and a time to overflight of 4.2 h.

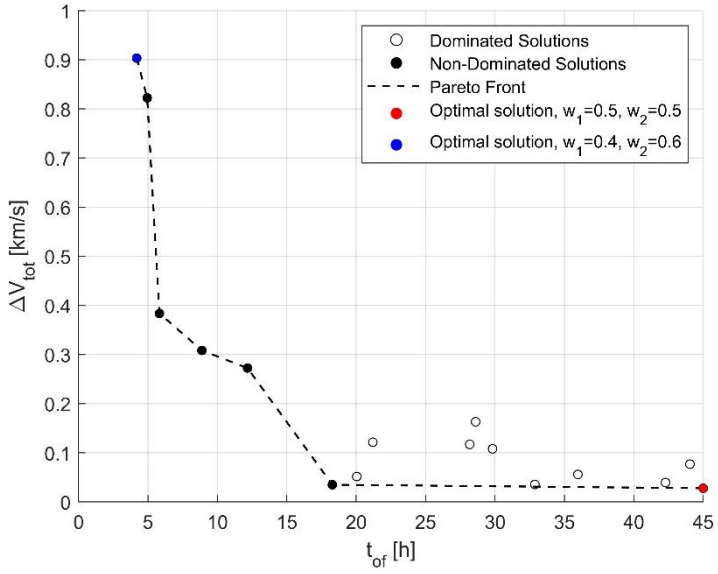


Figure 4.23 - Non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the simple additive weighting method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) with 3 satellites equipped with an optical sensor and 2 satellites equipped with a radar sensor. Same solution to the former for (w_1, w_2) equal to (0.6, 0.4).

Table 4.13 - Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
1	1 (D)	0.308	8.9	0.319	0.281	0.243
2	1 (A)	0.822	4.9	0.522	0.441	0.359
3	1 (A)	0.384	5.8	0.461	0.396	0.331
	1 (D)	0.0349	18.3	0.456	0.513	0.570
4	1 (A)	0.272	12.2	0.247	0.223	0.199
	2 (D)	0.0278	24 + 21.0	0.456	0.547	0.637
5	1 (D)	0.903	4.2	0.612	0.515	0.419

With regards to case B, the results of the proposed architecture are reported in Table 4.14. For the second, the third, the fourth and the fifth satellites, only ascending or descending passes have been considered depending on the illumination conditions. For the first satellite, instead, both ascending and descending passes have been taken into account since all the overflights take place during the daytime.

Table 4.14 - ΔV_{tot} and t_{of} for case B, considering $D = 2$. Only passages (ascending and/or descending) compatible with illumination conditions are reported.

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]
1	1 (D)	0.308	8.9
	1 (A)	0.0517	20.0
	2 (D)	0.0355	24 + 8.9
	2 (A)	0.0769	24 + 20.0
2	1 (A)	0.822	4.9
	2 (A)	0.163	24 + 4.6
3	1 (A)	0.497	5.6
	2 (A)	0.128	24 + 5.6
4	1 (A)	0.0859	21.5
	2 (A)	0.104	24 + 21.3
5	1 (D)	0.903	4.2
	2 (D)	0.117	24 + 4.2

Table 4.15 reports the score evaluation for the subset of non-dominated depicted in Figure 4.24, considering the three selected sets of weights. By assigning equal weight to ΔV_{tot} and t_{of}, the optimal solution corresponds to the observation occurring in the descending pass of the second overflight window by the first satellite, with a time to overflight of 32.9 h and ΔV_{tot} = 0.0355 km/s. The same solution is obtained for the pair of weights (0.6, 0.4). Increasing the weight assigned to t_{of} to 0.6, it is preferable to manoeuvre the fifth satellite, considering the descending pass of the first overflight window, with a time to overflight of 4.2 h and ΔV_{tot} = 0.903 km/s.

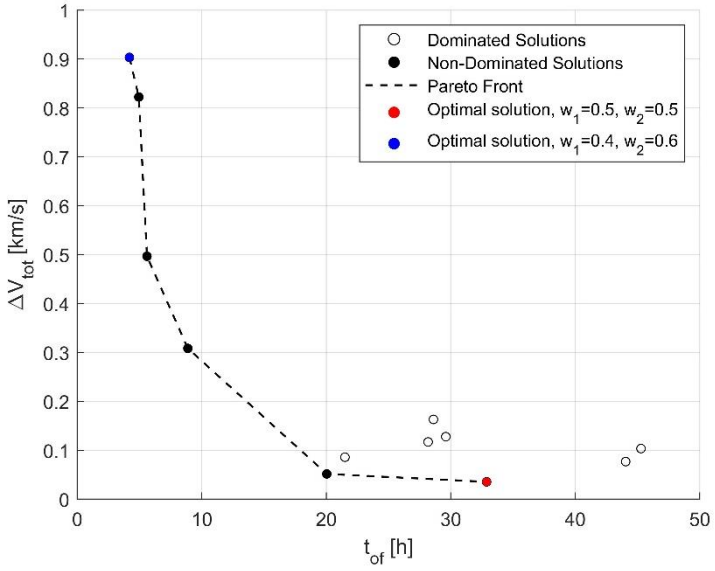


Figure 4.24 Non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the SAW method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) with all satellites equipped with an optical sensor. Same solution to the former for (w_1, w_2) equal to (0.6, 0.4).

Table 4.15 - Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
	1 (D)	0.308	8.9	0.329	0.293	0.258
1	1 (A)	0.0517	20.0	0.401	0.448	0.496
	2 (D)	0.0355	24 + 8.9	0.476	0.564	0.651
2	1 (A)	0.822	4.9	0.526	0.445	0.365

3	1 (A)	0.497	5.6	0.477	0.410	0.342
5	1 (D)	0.903	4.2	0.616	0.520	0.424

Table 4.16 provides the results of the proposed architecture for case C. Figure 4.25 shows the results of the two-objective optimization: namely (i.e., the dominated and non-dominated solutions and the Pareto frontier) as well as the optimal solution obtained by applying the SAW method for pairs of weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6). As can be seen in Figure 4.25, in the case under consideration, the Pareto frontier is composed of two non-dominated solutions: a manoeuvring and a non-manoeuvring overflight solution.

Table 4.17 reports the score evaluation for the subset of non-dominated solutions, considering the three selected sets of weights. By assigning equal weight to ΔV_{tot} and t_{of}, the best solution corresponds to the non-manoeuvring overflight that occurs in the ascending passage of the first overflight window by the fifth satellite, with a time to overflight of 15.5 h. Increasing the weight assigned to ΔV_{tot} to 0.6, the non-manoeuvring solution is still the best one. Increasing the weight assigned to t_{of} to 0.6, instead, it is advantageous to manoeuvre the fifth satellite, considering the descending pass of the first overflight window, with a time to overflight of 3.9 h and ΔV_{tot} = 0.136 km/s.

Table 4.16 - ΔV_{tot} and t_{of} for case C, considering $D = 2$. Only passages (ascending and/or descending) compatible with illumination conditions are reported.

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]
1	1 (D)	0.359	8.7
	1 (A)	0.0315	20.2
	2 (D)	0.0483	24 + 8.7
	2 (A)	0.0677	24 + 20.2
2	1 (A)	0.712	5.1
	1 (D)	0.341	11.7
	2 (A)	0.146	24 + 4.8
	2 (D)	0.128	24 + 11.4
3	1 (A)	0.384	5.8
	1 (D)	0.0349	18.3
	2 (A)	0.108	24 + 5.8

	2 (D)	0.0394	24 + 18.3
	1 (A)	0.272	12.2
4	1 (D)	0.122	21.2
	2 (A)	0.0558	24 + 12.0
	2 (D)	0.0278	24 + 21.0
	1 (D)	0.136	3.9
5	1 (A)	0	15.5
	2 (D)	0	24 + 3.9
	2 (A)	0.0930	24 + 15.4

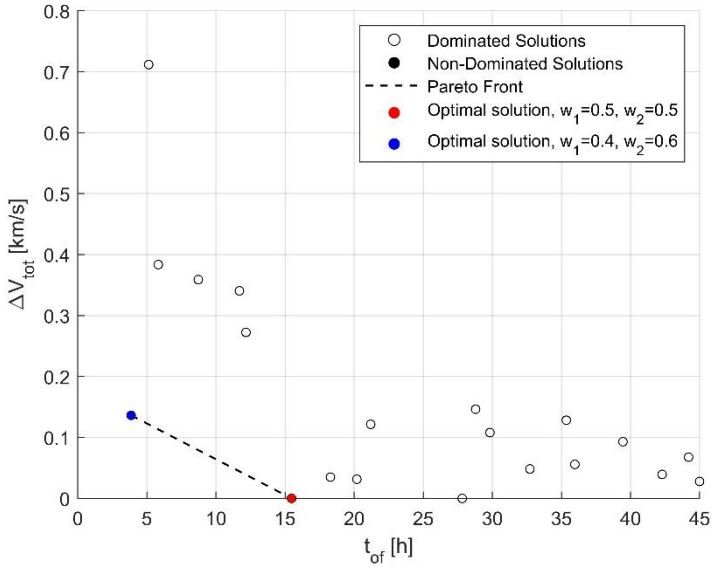


Figure 4.25 - Non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the SAW method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) with all satellites equipped with a radar. Same results as the former for pair of weights (0.6, 0.4).

Table 4.17 - Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
5	1 (D)	0.136	3.9	0.600	0.500	0.400
	1 (A)	0	15.5	0.550	0.625	0.700

To sum up, the following considerations can be made. By considering different payload configurations, the overflight solutions obtained from the overflight manoeuvre design block do not show any appreciable variations in ΔV_{tot} and t_{of}. However, the number of overflight solutions obtained in the scenario where all satellites are equipped with radar is always greater than or at most equal to the number of overflight solutions obtained by considering one or more satellites equipped with optical sensors. Indeed, for satellites equipped with an optical sensor, illumination conditions are taken into account and, thus, some overflight solutions are discarded depending on the latitude of the area of interest and on the selected initial epoch. Even though the number of overflight solutions given as input to the optimization algorithm changes depending on the payload configuration, the optimization results are not significantly affected by the type of on-board sensors.

4.2.4 Real test case

Finally, the proposed methodology is applied to a set of five real satellites for disaster monitoring. Table 4.18 provides the orbital parameters and the initial epoch of each satellite obtained from True Line Element (TLE) sets dated November 17, 2023 [167]. Since, each TLE set corresponds to a slightly different epoch, the satellites' trajectories are first propagated up to the latest one to obtain the initial conditions of the analysis. All the satellites fly on a sun-synchronous orbit. With the exception of RISAT-1 that has an on-board radar, all the other satellites are equipped with optical sensors. The specifications of the on-board sensors, listed in Table 4.19, are taken from [168], [169], [170], [171].

Table 4.18 - Orbital elements and epoch of the disaster monitoring satellite.

Satellite	a [km]	e [-]	i [°]	Ω [°]	ω [°]	ν [°]	$Epoch$
Deimos-1	7030.2	0.000197	97.68	149.94	40.30	40.18	17/11/2023 12:22:15.468
UK-DMC 2	7031.0	0.000156	97.752	113.37	82.76	82.64	17/11/2023 10:29:19.671
RISAT 1	6887.1	0.000140	97.54	333.87	268.36	91.77	17/11/2023 11:58:11.794
HJ-1B	7009.2	0.00397	97.64	338.56	32.90	32.78	17/11/2023 10:42:15.455
DMC3-FM2	6997.8	0.000623	97.78	202.27	273.90	86.22	17/11/2023 12:03:18.755

Table 4.19 - FOV along the cross-track α and tangential β directions for satellites equipped with an optical sensor; in-track angle γ, minimum, θ_{min}, and maximum, θ_{max}, off-nadir angles for satellite equipped with a radar.

	(α, β)	$(\gamma, \theta_{min}, \theta_{max})$
Deimos-1	(51.6°, 12.9°)	-
UK-DMC 2	(51.5°, 12.9°)	-
RISAT 1	-	(10°, 10°, 49°)
HJ-1B	(57.2°, 14.3°)	-
DMC3-FM2	(2.0°, 0.5°)	-

The AOI, shown in Figure 4.26, is Port-au-Prince in Haiti, which experienced a strong earthquake in 2010. The vertices of the rectangular AOI are: P_1 ($\lambda = -73.0°$, $\phi = 19.0°$), P_2 ($\lambda = -72.0°$, $\phi = 19.0°$), P_3 ($\lambda = -72.0°$, $\phi = 18.0°$) and P_4 ($\lambda = -73.0°$, $\phi = 18.0°$).

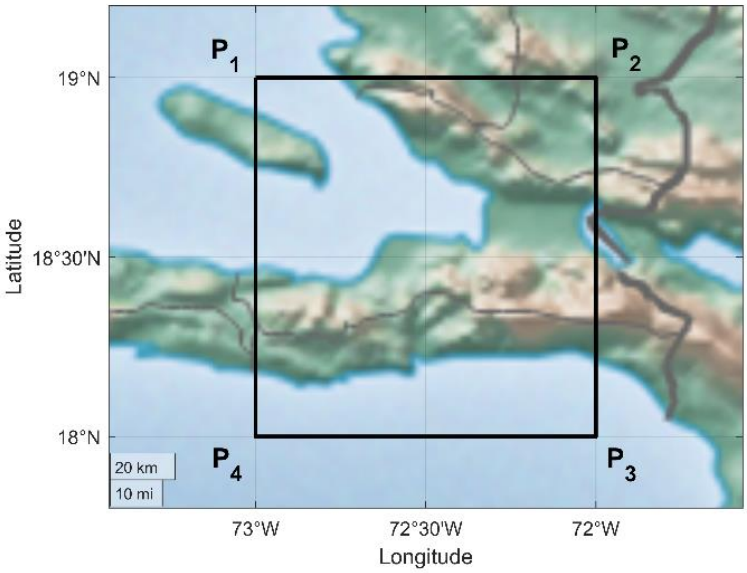


Figure 4.26 - AOI of the real test case scenario.

Table 4.20 shows the results for the set of disaster monitoring satellites. Since the first, second, third and fifth satellites are equipped with an optical sensor, only ascending or descending passes are considered, depending on the illumination conditions.

Table 4.20 - ΔV_{tot} and t_{of} for each satellite of the real scenario set. Only passages (ascending and/or descending) compatible with illumination conditions are reported.

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]
1	1 (A)	0	22.6
	2 (A)	0.0310	24 + 22.6
2	1 (D)	0.380	8.5
	2 (D)	0	24 + 8.5
3	1 (A)	0	11.1
	1 (D)	0	22.8
	2 (A)	0.00733	24 + 10.9
	2 (D)	0.0585	24 + 23.0
4	1 (D)	0.140	23.5
	2 (D)	0	24 + 23.6
5	1 (A)	1.247	2.0
	2 (A)	0.116	24 + 2.0

Figure 4.27 shows the results of the two-objective optimization: namely (i.e., the dominated and non-dominated solutions and the Pareto frontier) as well as the optimal solution obtained by applying the SAW method for pairs of weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6). As can be seen in Figure 4.27, there is a non-manoeuvring overflight that lies on the Pareto frontier. The score evaluation for the subset of non-dominated solutions is reported in Table 4.21, considering the three selected sets of weights. By assigning equal weight to ΔV_{tot} and t_{of}, the optimal solution corresponds to the non-manoeuvring overflight occurring in the ascending pass of the first overflight window by the third satellite, with a t_{of} = 11.1 h. Increasing the weight assigned to the ΔV_{tot} to 0.6, the non-manoeuvring solution is still the best one. Instead, when assigning more weight to the time to overflight (w_2=0.6), it is preferable to manoeuvre the fifth satellite considering the ascending pass of the first overflight window, with ΔV_{tot} = 1.247 km/s and t_{of} = 2.0 h.

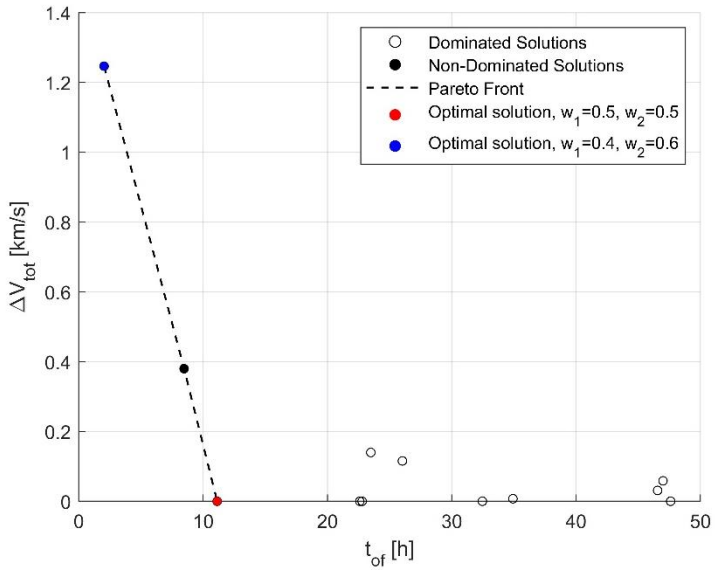
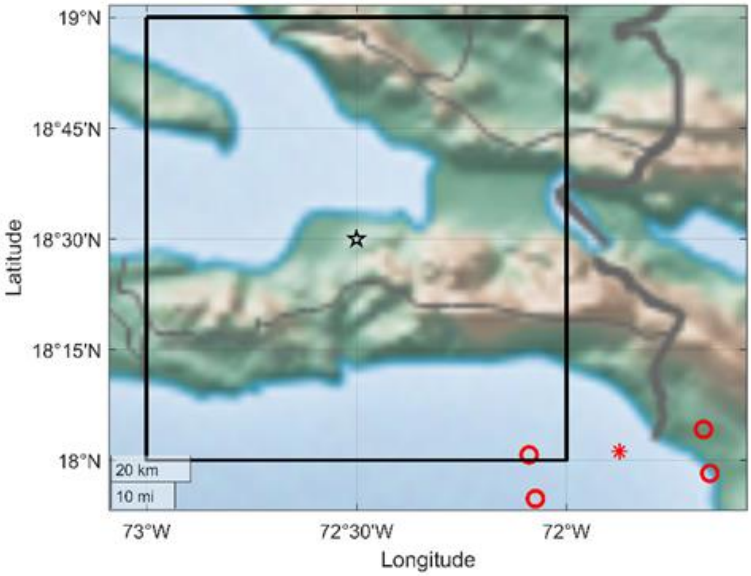


Figure 4.27 - Dominated solutions, non-dominated solutions, Pareto frontier and the optimal solution obtained by applying the SAW method with weights (w_1, w_2) equal to (0.5, 0.5) and (0.4, 0.6) for t_{of} = 2 days in a real test case scenario. Same results as the former for (0.6, 0.4).

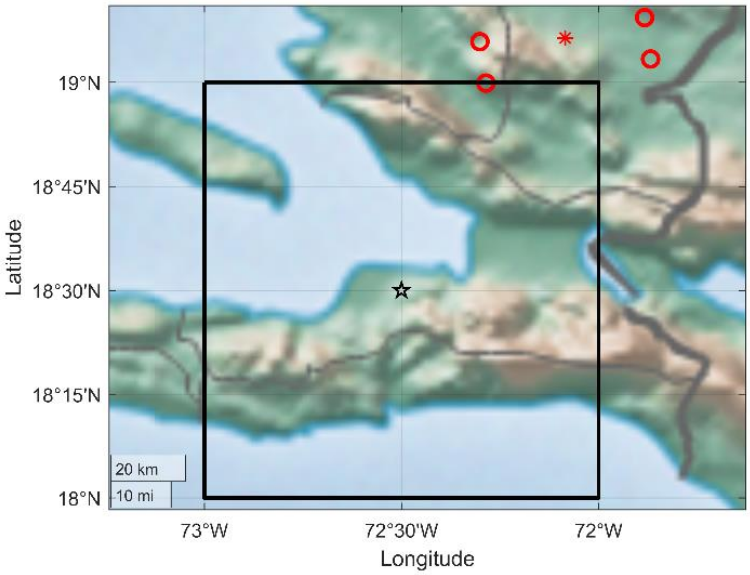
Table 4.21 - Performance score for each non-dominated solution obtained by applying the SAW method for weights pair (w_1, w_2) equal to (0.4, 0.6), (0.5, 0.5) and (0.4, 0.6).

Satellite #	Day (A/D)	ΔV_{tot} [km/s]	t_{of} [h]	Score		
				w_1=0.4 w_2=0.6	w_1=0.5 w_2=0.5	w_1=0.6 w_2=0.4
2	1 (D)	0.380	8.5	0.144	0.120	0.0962
3	1 (A)	0	11.1	0.510	0.592	0.673
5	1 (A)	1.247	2.0	0.600	0.500	0.400

Figure 4.28 illustrates the observation made by the fifth satellite considering the ascending pass of the first overflight window. Specifically, the sensor footprint when the satellite enters and exits the AOI is depicted in Figure 4.28 (a) and (b) respectively. As shown in Figure 4.28, the footprint of the manoeuvred satellite intersects the AOI and, thus, it is observable. However, the barycentre of the footprint does not have the same coordinates as the barycentre of the AOI. This is due to the approximations of the analytical model (which does not account for orbital perturbations other than J_2) that could be taken into account by further shifting the barycentre of the footprint of a certain quantity.



(a)



(b)

Figure 4.28 - Footprint of the manoeuvred satellite at the entrance (a) and exit point (b) of the area of interest for (w_1, w_2) equal to (0.4, 0.6). The empty dots are the vertices of the footprint and the asterisk is the sub-satellite point. Black lines mark the area of interest, and the black star represents the barycentre of the area of interest.

5. Conclusions and future developments

The research activity presented in this thesis was addressed to the development and performance assessment of innovative approaches and techniques to support advanced functions for Space Situational Awareness.

In the framework of fragmentation events characterization, an iterative approach was proposed to estimate the tuning parameters of the Standard Breakup Model, namely, the fragmented mass of the parent objects in case of collision events. A sensitivity analysis was performed and demonstrated that the parents' masses do not affect the velocity distribution of the fragments. Therefore, the tuning of the masses cannot be based on the orbital parameters of the fragments but rather on the number of catalogued objects. The catalogued objects are backward propagated to the fragmentation event epoch and separated into two clusters corresponding to the target and projectile debris clouds. This allows performing the iterative approach on the two parent objects separately and, hence, estimating their fragmented masses. The approach was applied to three test cases, namely two simulated collisions (one catastrophic, and one non-catastrophic), and a real fragmentation event (on-orbit collision between DMSP 5B F5 Thor Burner and the Catalogued debris 26207). In all the cases, the algorithm has converged and provided reasonable values of the tuning parameters, obtaining a number of fragments equal, within a certain tolerance, to the number of catalogued objects. In particular, for the non-catastrophic collision and the real fragmentation event, big discrepancies were detected between the fragmented mass and number of fragments estimated by the SBM with nominal parameters, and the same outputs obtained by the iterative approach, with the latter being much closer to expected and real data. This also means that the tuned SBM can be used to better assess the number and distributions of non-detectable fragments generated by the collision event.

In the context of Resident Space Objects characterization, an algorithm for estimating the ballistic coefficient of RSOs in LEO was proposed, using Machine Learning based techniques. Training, validation and testing data sets have been

generated extracting synthetic space objects from pre-defined intervals of area-to-mass ratio and orbital elements. Three different types of ML-techniques, available on MATLAB Regression Learner App, have been compared considering different combinations of possible features, using both osculating and mean orbital elements. The RMSE has been chosen as the performance metric, which has led to a specific type of ML technique (i.e., neural networks), a specific set of features (i.e., OEl_0, δOEl, and Δt), and the choice of mean orbital elements for the subsequent analyses. A sensitivity analysis has been carried out evaluating the behaviour of the neural networks with respect to the number of trainings, the propagation time, the number of objects of the training data set, and the frequency of the measurements. As a consequence, the original training data set has been modified. Then, the algorithm has been applied to the testing set estimating the *BC* for each object with all the neural networks. The testing set objects have been propagated with the predicted values of the *BC* and the semimajor axis thus obtained has been compared with the true one. The TNN is the neural network that exhibits the highest percentage of success, equal to 84%. Moreover, the *BC* for most of the objects can be properly predicted by all the neural networks and, setting a threshold of 5 km on the difference between the true and predicted semimajor axis, all the neural networks have a percentage of success at least equal to 80%. Two types of objects, instead, are not characterised correctly: objects that either re-enter the atmosphere or that are characterized by a very low *BC*. In both cases, a longer observability period and/or a wider training set would be necessary. Finally, the applicability of the algorithm to real space objects has been evaluated considering 10 debris taken from a space catalogue and two active satellites of Lemur-2 constellation. For each RSO, the ballistic coefficient has been estimated with every neural network and the object trajectory has been, then, propagated. For 9 debris, there is at least one neural network that leads to a maximum error on the semimajor axis lower than 1 km after 30 days of propagation, despite the uncertainties on the atmospheric model, on the C_d, and the model used to convert the orbital elements from osculating to mean. For the Lemur-2 constellation satellites, all the neural networks lead to a Δa_{max} lower than 2 km and the MNN and WNN lead to Δa_{max} lower than 1 km. Results on real objects have thus demonstrated that the algorithm can be generally applied.

A further analysis has aimed to include the effects of the solar activity in the training set, adding data corresponding to a period of the year with a higher average value of K_p, and a binary feature that corresponds to a low or high value of the K_p index. Applying the new trained network to the testing set, it has been demonstrated that the solar activity must be included in the training set when the testing set refers to a different period of the year.

The NNs have been compared with a semi-analytical method in terms of performance and computational cost across the testing dataset. The WNN has exhibited the lowest median of the Err_{BC} (absolute value of the percentage error) distribution, and all the NNs have a median of Err_{BC} distribution lower than the SAM. On the other hand, the SAM has exhibited the lowest standard deviation. Furthermore, when comparing the NNs with the SAM on real test cases, the latter exhibits a Δa_{max} similar to the one of the NN.

Starting from the proposed methodology and the results so far obtained, two additional contributions have been explored and analysed. First, a reduced set of features to train the Neural Networks was proposed. In particular, since the two orbital parameters mostly influenced by the *BC* are the semimajor axis and the eccentricity, only the consistency errors δa and δe have been considered, instead of *δOEl*. For the TNN the error metrics improve with equal computational cost. For the WNN instead, there is a decrease of 17% of the computational cost and a slight reduction of the mean percentage error. This means that the proposed set of features is able to capture the main effects of the *BC* on the object trajectory. Then, two data preprocessing techniques have been explored: normalization and moving mean. Normalizing the data has led to a decrease of both the mean and the median of the percentage error for high-BC objects, whereas no significant improvement has been observed for the low-BC and the medium-BC subsets. Moreover, the computational cost has decreased of about 15% for the two networks. As regards the implementation of a moving mean on the input data, the performance has been assessed with respect to the number of points over which the mean is computed (k). In all the cases, there is at least one value of k that leads to an improvement of the performance metrics with respect to the case where no moving mean is applied. In particular, for the low-BC objects, whose mean orbital

parameters exhibit larger oscillations, the mean percentage error decreases of about 25 percentage points.

Another important aspect of this thesis has regarded analysing the capability of the proposed ML algorithm to detect a *BC* variation. A step function for the *BC* has been assumed and the performance of the networks has been investigated with respect to the percentage of variation pc and the time at which such a variation occurs t_1. For pc equal to +100%, for every considered value of t_1, the estimated *BC* starts to increase for $t > t_1$ and gets closer to the reference value. In particular, for t_1= 30 days, the error of the predicted *BC* is lower than 3% on the last propagation day. For pc equal to +50%, similar results have been obtained, clearly with an increase the in the percentage error. Such a behaviours can be seen for objects with a $0.2 < BC_0 < 0.65$ m^2/kg. On the contrary, for $BC_0 < 0.2$ m^2/kg, two trends have been observed: either the *BC* starts to increase at a time much larger than t_1 (e.g., after 90 days) or the neural network does not detect any variation of the *BC* in the observation window.

Besides Feedforward Neural Networks, another architecture has been explored, namely, Recurrent Neural Network. In particular, a GRU-based neural network was proposed to estimate the ballistic coefficient of Resident Space Objects in LEO, exploiting the time evolution of the mean orbital parameters over a time period of 180 days (or until re-entry in atmosphere). Similarly to the FNN case, training, validation, and testing data sets have been generated by randomly sampling initial conditions corresponding to synthetic space objects from pre-defined intervals of area-to-mass ratio and orbital elements. A high-accuracy numerical propagator has been used for the generation of the synthetic trajectories. The obtained data have been normalised and "padded". The designed neural network includes two feedforward and one GRU layer. Several hyperparameters have been tuned. In particular, appropriate tuning of the maximum number of training epochs, the batch size and the initial learning rate have been demonstrated to reduce training and validation RMSE up to 50%. The performance of the trained network has been evaluated on a set of numerical test cases. Assuming an exponential atmospheric model, the performance metrics μ, m, and σ are 9.9%, 3.7% and 20.0%. Afterwards, the impact of the atmospheric density variability and uncertainty has been assessed by increasing the complexity of the atmospheric

model. The NRLMSISE-00 model, first with fixed space weather coefficients and then with time-varying coefficients, has been considered. In all the cases, the performance metrics μ and m are smaller than 11% and 7% respectively. These results are comparable with the ones obtained using only FNNs with the consistency error approach. However, the GRU-based network needs half of the computational time to generate the training dataset. Moreover, dividing the testing dataset in three subintervals (i.e., low, medium and high-BC), it has been noticed that for medium and high-BC objects (which represent 94% of the testing dataset), μ and m assume values one order of magnitude smaller than the low-BC subset. Considering the NRLMSISE-00 model with time-varying coefficients, two additional results have been shown. First, the mean percentage error versus elapsed time is analysed. The RNN needs 4 months to reach the final predicted value, whereas the WNN needs 30 days. This might mean that the time evolution of the consistency errors has a clearer trend than the time evolution of the orbital parameters. Second, the testing set objects have been propagated with the predicted values of the *BC* and the semimajor axis thus obtained has been compared with the true one. Up to 30 days, 90% of the testing dataset has a Δa smaller than 5 km. This percentage decreases to 68% after 180 days. Sensitivity with respect to atmospheric model uncertainty has been analysed. In particular, propagating the training dataset objects with the SISE model and the testing dataset objects with the exponential model, thus having up to 100% on local variations of density values, the performance metrics μ and m are smaller than 15% and 10% respectively. Sensitivity analysis in terms of the number of objects of the training dataset has also been carried out. If this number is moved from 3000 to 5000, the performance metrics get slightly worse for the exponential model, facing overfitting. Vice versa, for a more complex model such as the NRLMSISE-00, the use of a larger dataset improves the performance metrics of a few percentage points. The proposed algorithm has been generalised also to unevenly spaced input data. In particular, for a minimum mean sampling frequency of 2 measurements per day, the performance is comparable with evenly spaced data. Finally, the algorithm has been demonstrated to be robust with respect to orbit determination uncertainty. This is a notable advantage when compared to the consistency-error based approach, since an initial orbit determination error is not propagated. This work has proven that supervised recurrent

neural networks of modest complexity are a viable option for the rapid characterisation of the ballistic coefficients of large sets of objects.

To bring both approaches (i.e., purely feedforward and GRU-based NNs) closer to operational capability, the next steps should be focused along two directions. The first concerns making the present method more robust with respect to time-varying space weather conditions by including driving indices, such as $F_{10.7}$, within the feature vector, as preliminary demonstrated for the FNNs. The aim is to enable the network to decouple atmospheric and ballistic coefficient uncertainties during training, building on encouraging recent results in employing machine learning methods to augment existing high-fidelity atmospheric models [172], [173], [174], [175]. The second concerns the quantification of the uncertainty associated to the prediction of the *BC*, which will enable the present approach to be robustly implemented within existing SSA systems. This can be achieved through several approaches, such as Bayesian extensions to recurrent neural network models [176]. Once these directions have been investigated appropriately, the resulting NNs can be further tested on actual astrometric data, such as Two-Line Element time series. Further aspects to be investigated include the estimation of time-varying *BC*s according to more complex functions, along with radiation *BC*s in MEO and GEO orbital regimes, and using NNs in conjunction with a coarse classification step to improve estimation accuracy for very high or low *BC* values.

In the framework of mission characterization several tools have been developed to support RSP functions, such as visibility analysis from ground-based sensors and GNSS coverage analysis over AOIs. Moreover, to support responsive space capabilities, it was proposed a decision-making architecture to select the optimal manoeuvre that enables a set of multiple sensor-equipped satellites to overfly a specified area of interest within a certain maximum time to the expected overflight. This architecture is composed of three main blocks. The overflight block enables to check whether any satellites overfly the AOI without the need of manoeuvring within the required time. For the subset of satellites that do not overfly the area of interest, the manoeuvre design block is activated. A dual coplanar impulsive manoeuvre is designed by modifying the analytical J2-perturbed model proposed by Zhang et al. [9]

to account for on-board sensors (i.e., optical and radar) and an extended (non-punctual) AOI. Once the overflight solutions in terms of ΔV_{tot} (equal to 0 km/s for non-manoeuvring satellites) and time to overflight are computed for each satellite of the initial set, an optimization block is implemented to determine which satellite to manoeuvre. A multi-objective optimization problem is solved to find a set of equally optimal solutions (i.e., non-dominated solutions) that form the Pareto optimal front. Then, the simple additive weighting (SAW) method is applied to select the optimal solution according to the decision-making preferences. The decision-making architecture has been validated by considering both simulated and real test cases. In the former case, different scenarios have been simulated varying the type of on-board sensors and the area of interest. In the latter case, a set of real Earth-observation satellites has been considered. In both cases, the proposed architecture outputs an optimal solution.

It should be noted that the proposed architecture does not consider attitude or pointing manoeuvres to optimize coverage of an area of interest. In addition, the manoeuvre design block only takes into account the J_2 perturbation, neglecting the aerodynamic drag, which plays an important role in LEO orbit. Therefore, future developments could introduce the possibility of performing attitude or pointing manoeuvres to provide a better view of the Earth site; a ΔV adjustment in the manoeuvre design block to account for the effect of the aerodynamic drag; low-thrust manoeuvres instead of impulsive manoeuvres; and the overflight quality as an additional parameter to be accounted for in the optimization algorithm.

Bibliography

[1] L. Miraux, “Environmental limits to the space sector’s growth,” *Science of the Total Environment*. Elsevier B.V., 2021. doi: 10.1016/j.scitotenv.2021.150862.

[2] I. Gkolias, J. Daquin, F. Gachet, and A. J. Rosengren, “FROM ORDER TO CHAOS IN EARTH SATELLITE ORBITS,” *Astron J*, vol. 152, no. 5, 2016, doi: 10.3847/0004-6256/152/5/119.

[3] J. Daquin, A. J. Rosengren, E. M. Alessi, F. Deleflie, G. B. Valsecchi, and A. Rossi, “The dynamical structure of the MEO region: long-term stability, chaos, and transport,” *Celest Mech Dyn Astron*, vol. 124, no. 4, 2016, doi: 10.1007/s10569-015-9665-9.

[4] European Space Agency, “ESA’s Annual Space Environment Report,” Sep. 2023.

[5] M. J. Holzinger and M. K. Jah, “Challenges and potential in space domain awareness,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 1, 2018, doi: 10.2514/1.G003483.

[6] T. Donath, T. Schildknecht, V. Martinot, and L. Del Monte, “Possible European systems for space situational awareness,” *Acta Astronaut*, vol. 66, no. 9–10, 2010, doi: 10.1016/j.actaastro.2009.10.036.

[7] United Nations, “Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space,” *United Nations*, 2010.

[8] J. A. Kennewell and B. N. Vo, “An overview of space situational awareness,” in *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013*, 2013.

[9] Y. Wang and M. Vasile, “Intelligent selection of NEO deflection strategies under uncertainty,” *Advances in Space Research*, vol. 72, no. 7, 2023, doi: 10.1016/j.asr.2022.08.086.

[10] L. Walker and M. Vasile, “Space debris remediation using space-based lasers,” *Advances in Space Research*, vol. 72, no. 7, 2023, doi: 10.1016/j.asr.2023.06.031.

[11] E. M. Alessi *et al.*, “Effectiveness of GNSS disposal strategies,” *Acta Astronaut*, vol. 99, no. 1, 2014, doi: 10.1016/j.actaastro.2014.03.009.

[12] M. Poozhiyil *et al.*, “Active debris removal: A review and case study on LEOPARD Phase 0-A mission,” *Advances in Space Research*, vol. 72, no. 8, 2023, doi: 10.1016/j.asr.2023.06.015.

[13] M. C. Wijayatunga, R. Armellin, H. Holt, L. Pirovano, and A. A. Lidtke, “Design and guidance of a multi-active debris removal mission,” *Astrodynamics*, vol. 7, no. 4, 2023, doi: 10.1007/s42064-023-0159-3.

[14] Y. Xu, X. Liu, R. He, Y. Zhu, Y. Zuo, and L. He, “Active Debris Removal Mission Planning Method Based on Machine Learning,” *Mathematics*, vol. 11, no. 6, 2023, doi: 10.3390/math11061419.

[15] Y. LIU *et al.*, “The consequences of Cosmos 1408 breakup: The optimization of the breakup model and the influence of debris clouds,” *Acta Astronaut*, vol. 206, 2023, doi: 10.1016/j.actaastro.2023.02.023.

[16] S. Flegel, J. Bennett, M. Lachut, M. Möckel, and C. Smith, “An analysis of the 2016 Hitomi breakup event,” *Earth, Planets and Space*, vol. 69, no. 1, 2017, doi: 10.1186/s40623-017-0633-3.

[17] D. J. Lee, E. J. Choi, S. Cho, J. H. Jo, and T. S. No, “Effective Computational Approach for Prediction and Estimation of Space Object Breakup Dispersion during Uncontrolled Reentry,” *International Journal of Aerospace Engineering*, vol. 2018, 2018, doi: 10.1155/2018/6824978.

[18] H. Cowardin *et al.*, “Updates on the DebriSat hypervelocity experiment and characterization of fragments in support of environmental models,” *Int J Impact Eng*, vol. 180, 2023, doi: 10.1016/j.ijimpeng.2023.104669.

[19] Y. Zhang, F. An, S. Liao, C. Wu, J. Liu, and Y. Li, “Study on numerical simulation methods for hypervelocity impact on large-scale complex spacecraft structures,” *Aerospace*, vol. 9, no. 1, 2022, doi: 10.3390/aerospace9010012.

[20] D. L. Mains and M. E. Sorge, “The IMPACT satellite fragmentation model,” *Acta Astronaut*, vol. 195, 2022, doi: 10.1016/j.actaastro.2022.03.030.

[21] N. L. Johnson, P. H. Krisko, J. C. Liou, and P. D. Anz-Meador, “NASA’s new breakup model of EVOLVE 4.0,” *Advances in Space Research*, vol. 28, no. 9, 2001, doi: 10.1016/S0273-1177(01)00423-9.

[22] R. Linares and J. L. Crassidis, “AAS 16-514 RESIDENT SPACE OBJECT SHAPE INVERSION VIA ADAPTIVE HAMILTONIAN MARKOV CHAIN MONTE CARLO,” 2016.

[23] R. Linares, M. K. Jah, and J. L. Crassidis, “SPACE OBJECT AREA-TO-MASS RATIO ESTIMATION USING MULTIPLE MODEL APPROACHES,” 2012.

[24] G. D. Badhwar and P. D. Anz-Meador, “Determination of the Area and Mass distrubution of orbital debris fragments,” *Earth Moon Planets*, pp. 29–51, 1989.

[25] E. Lacruz, D. Casanova, and A. Abad, “Estimation of a reliability range for the area-to-mass ratio of orbiters at the geostationary ring,” *Acta Astronaut*, vol. 166, pp. 104–112, Jan. 2020, doi: 10.1016/j.actaastro.2019.09.039.

[26] V. Ray and D. J. Scheeres, “Drag coefficient model to track variations due to attitude and orbital motion,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 10, 2020, doi: 10.2514/1.G004854.

[27] W. K. Tobiska *et al.*, “The SET HASDM Density Database,” *Space Weather*, vol. 19, no. 4, 2021, doi: 10.1029/2020SW002682.

[28] R. Linares and R. Furfaro, “Space Object classification using deep Convolutional Neural Networks; Space Object classification using deep Convolutional Neural Networks,” 2016.

[29] R. Furfaro, R. Linares, D. Gaylor, M. Jah, and R. Walls, “Resident Space Object Characterization and Behavior Understanding via Machine Learning and Ontology-based Bayesian Networks,” in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2016. [Online]. Available: www.amostech.com

[30] H. Peng and X. Bai, “Artificial neural network-based machine learning approach to improve orbit prediction accuracy,” in *Journal of Spacecraft and Rockets*, American Institute of Aeronautics and Astronautics Inc., 2018, pp. 1248–1260. doi: 10.2514/1.A34171.

[31] M. Khalil, E. Fantino, and P. Liatsis, “Classification of space objects using machine learning methods,” in *Proceedings - 2019 IEEE 1st International Conference on Cognitive Machine Intelligence, CogMI 2019*, Institute of Electrical and Electronics Engineers Inc., Dec. 2019, pp. 93–96. doi: 10.1109/CogMI48466.2019.00021.

[32] R. Abay, S. Balage, M. Brown, and R. Boyce, “Two-Line Element Estimation Using Machine Learning,” *Journal of the Astronautical Sciences*, vol. 68, no. 1, pp. 273–299, Mar. 2021, doi: 10.1007/s40295-021-00249-0.

[33] R. Mital, K. Cates, J. Coughlin, and G. Ganji, “A Machine Learning Approach to Modeling Satellite Behavior,” in *Proceedings - 2019 IEEE International Conference on Space Mission Challenges for Information Technology, SMC-IT 2019*, Institute of Electrical and Electronics Engineers Inc., Jul. 2019, pp. 62–69. doi: 10.1109/SMC-IT.2019.00013.

[34] H. Kato *et al.*, “Machine Learning for Atmospheric Drag Prediction of LEO satellites,” 2018.

[35] B. Li, J. Huang, Y. Feng, F. Wang, and J. Sang, “A Machine Learning-Based Approach for Improved Orbit Predictions of LEO Space Debris with Sparse Tracking Data from a Single Station,” *IEEE Trans Aerosp Electron Syst*, vol. 56, no. 6, pp. 4253–4268, Dec. 2020, doi: 10.1109/TAES.2020.2989067.

[36] C. A. Goodfellow lan, Bengio Yoshua, “Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville - Google Books,” *MIT Press*. 2016.

[37] SatCen, “The Recognized Space Picture (RSP) Display study.,” https://www.satcen.europa.eu/page/the-recognized-space-picture-rsp-display-study.

[38] Z. Szajnfarber, M. G. Richards, and A. L. Weigel, “Implications of DoD acquisition policy for innovation: The case of operationally responsive space,” in *Space 2008 Conference*, 2008. doi: 10.2514/6.2008-7650.

[39] T. C. Co and J. T. Black, “Responsiveness in low orbits using electric propulsion,” in *Journal of Spacecraft and Rockets*, 2014. doi: 10.2514/1.A32405.

[40] D. Mehrholz, L. Leushacke, W. Flury, R. Jehn, H. Klinkrad, and M. Landgraf, “Detecting, tracking and imaging space debris,” *ESA Bulletin(0376-4265)*, vol. 109, no. 109, 2002.

[41] H. G. Lewis, “Understanding long-term orbital debris population dynamics,” *Journal of Space Safety Engineering*, vol. 7, no. 3, 2020, doi: 10.1016/j.jsse.2020.06.006.

[42] V. Braun, S. Lemmens, B. Reihs, H. Krag, and A. Horstmann, “Analysis of breakup events,” *7th European Conference on Space Debris*, no. April, 2017.

[43] A. Rossi, L. Anselmo, C. Pardini, R. Jehn, and G. B. Valsecchi, “The new space debris mitigation (SDM 4.0) long term evolution code,” in *European Space Agency, (Special Publication) ESA SP*, 2009.

[44] N. L. Johnson, “Space debris modeling at NASA,” in *European Space Agency, (Special Publication) ESA SP*, 2001.

[45] P. H. Krisko, “The new NASA orbital debris engineering model ORDEM 3.0,” in *AIAA/AAS Astrodynamics Specialist Conference 2014*, 2014.

[46] P. H. Krisko and D. T. Hall, “Geosynchronous region orbital debris modeling with GEO_EVOLVE 2.0,” in *Advances in Space Research*, 2004. doi: 10.1016/j.asr.2003.10.040.

[47] P. H. Krisko and J. C. Liou, “NASA long-term orbital Debris modeling comparison: LEGEND AND EVOLVE,” in *54th International Astronautical Congress of the International Astronautical Federation (IAF), the International Academy of Astronautics and the International Institute of Space Law*, 2003.

[48] J. C. Liou and N. L. Johnson, “Instability of the present LEO satellite populations,” *Advances in Space Research*, vol. 41, no. 7, 2008, doi: 10.1016/j.asr.2007.04.081.

[49] A. E. White and H. G. Lewis, “An adaptive strategy for active debris removal,” *Advances in Space Research*, vol. 53, no. 8, 2014, doi: 10.1016/j.asr.2014.01.021.

[50] S. Allen and N. Fitz-Coy, “DebriSat fragment characterization: Quality assurance,” *Journal of Space Safety Engineering*, vol. 7, no. 3, 2020, doi: 10.1016/j.jsse.2020.08.001.

[51] G. Fonder, M. Hughes, M. Dickson, M. Schoenfeld, and J. Gardner, “Space Fence Radar Overview,” in *2019 International Applied Computational Electromagnetics Society Symposium in Miami, ACES-Miami 2019*, 2019.

[52] N. Griffith, E. Lu, M. Nicolls, I. Park, and C. Rosner, “Commercial Space Tracking Services for Small Satellites,” *Small Satellite Conference*, 2019.

[53] G. Muntoni, G. Montisci, T. Pisanu, P. Andronico, and G. Valente, “Crowded space: A review on radar measurements for space debris monitoring and tracking,” *Applied Sciences (Switzerland)*, vol. 11, no. 4. 2021. doi: 10.3390/app11041364.

[54] S. Diserens, H. G. Lewis, and J. Fliege, “NewSpace and its implications for space debris models,” *Journal of Space Safety Engineering*, vol. 7, no. 4, 2020, doi: 10.1016/j.jsse.2020.07.027.

[55] J. Murray, H. Cowardin, J.-C. Liou, M. Sorge, N. Fitz-Coy, and T. Huynh, “Analysis of the DebriSat Fragments and Comparison to the NASA Standard Satellite Breakup Model,” in *International Orbital Debris Conference,* 2019.

[56] M. Rivero *et al.*, “I AC-15-A6.2.9x30343 Characterization of debris from the debrisat hypervelocity test,” in *Proceedings of the International Astronautical Congress, IAC*, 2015.

[57] D. Mains and M. Sorge, “Integrated breakup modeling solutions from debrisat analysis,” *Journal of Space Safety Engineering*, vol. 7, no. 3, 2020, doi: 10.1016/j.jsse.2020.07.024.

[58] J. Murakami, T. Hanada, J. C. Liou, and E. Stansbery, “Micro-satellite impact tests to investigate multi-layer insulation fragments,” in *European Space Agency, (Special Publication) ESA SP*, 2009.

[59] T. Hanada, J. C. Liou, T. Nakajima, and E. Stansbery, “Outcome of recent satellite impact experiments,” *Advances in Space Research*, vol. 44, no. 5, 2009, doi: 10.1016/j.asr.2009.04.016.

[60] F. Hoots and M. Sorge, “Satellite breakup parameter determination,” in *Journal of the Astronautical Sciences,* 2012. doi: 10.1007/s40295-013-0009-2.

[61] P. D. Anz-Meador and M. J. Matney, “An assessment of the NASA explosion fragmentation model to 1 mm characteristic sizes,” in *Advances in Space Research*, 2004. doi: 10.1016/j.asr.2003.11.008.

[62] D. A. Vallado and D. L. Oltrogge, “FRAGMENTATION EVENT DEBRIS FIELD EVOLUTION USING 3D VOLUMETRIC RISK ASSESSMENT,” in *7th European Conference on Space Debris,* 2017.

[63] C. Pardini and L. Anselmo, “Review of past on-orbit collisions among cataloged objects and examination of the catastrophic fragmentation concept,” *Acta Astronaut,* vol. 100, no. 1, pp. 30–39, 2014, doi: 10.1016/J.ACTAASTRO.2014.03.013.

[64] F. Letizia, C. Colombo, H. Lewis, and C. R. McInnes, “Space debris cloud evolution in Low Earth Orbit,” in *Proceedings of the International Astronautical Congress, IAC*, 2013.

[65] S. Frey, C. Colombo, and S. Lemmens, “Application of density-based propagation to fragment clouds using the Starling suite,” in *1st International Space Debris Conference*, 2019.

[66] K. Tetreault, S. D. Ross, K. Schroeder, and J. Black, “Fragmentation event identification using back propagation with variable ballistic coefficient calculation,” in *Proceedings of the 19th annual Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Maui, HI, USA*, 2018.

[67] L. Dimare, S. Cicalò, A. Rossi, E. M. Alessi, and G. B. Valsecchi, “In-orbit fragmentation characterization and parent bodies identification by means of orbital distances.,” in *1st International Orbital Debris Conference (IOC)*, Sugarland, TX, USA, Dec. 2019.

[68] A. Muciaccia, M. Romano, and C. Colombo, “Detection and characterisation of in-orbit fragmentations over short and long periods of time,” in *Proceedings of the International Astronautical Congress, IAC*, 2021.

[69] J. Geul, E. Mooij, and R. Noomen, “TLE uncertainty estimation using robust weighted differencing,” *Advances in Space Research*, vol. 59, no. 10, 2017, doi: 10.1016/j.asr.2017.02.038.

[70] A. D. Dianetti and J. L. Crassidis, “Space Object Material Determination from Polarized Light Curves,” in *AIAA Scitech Forum*, 2019.

[71] J. Šilha, M. Zigo, T. Hrobár, P. Jevčák, and M. Verešvárska, “LIGHT CURVES APPLICATION TO SPACE DEBRIS CHARACTERIZATION AND CLASSIFICATION.” [Online]. Available: http://conference.sdo.esoc.esa.int,

[72] R. Linares, M. K. Jah, J. L. Crassidis, F. A. Leve, and T. Kelecy, “Astrometric and photometric data fusion for inactive space object mass and area estimation,”

Acta Astronaut, vol. 99, no. 1, pp. 1–15, Jun. 2014, doi: 10.1016/j.actaastro.2013.10.018.

[73] R. Linares, M. K. Jah, J. L. Crassidis, and C. K. Nebelecky, "Space Object Shape Characterization and Tracking Using Light Curve and Angles Data," 2014.

[74] D. N. Brack and P. Gurfil, "In-orbit tracking of high area-to-mass ratio space objects," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2030–2041, 2017, doi: 10.2514/1.G002501.

[75] "CelesTrak: EOP and Space Weather Data." Accessed: Feb. 18, 2022. [Online]. Available: https://celestrak.com/SpaceData/

[76] J. Carlos Dolado Pérez, L. Aivar Garcia, A. Águeda Maté, and I. Llamas de la Sierra, "OPERA: A tool for lifetime prediction based on orbit determination from TLE data," 2014.

[77] A. Águeda, L. Aivar, J. Tirado, and J. C. Dolado, "IN-ORBIT LIFETIME PREDICTION FOR LEO AND HEO BASED ON ORBIT DETERMINATION FROM TLE DATA," 2013.

[78] M. D. Pilinski *et al.*, "Comparative analysis of satellite aerodynamics and its application to space-object identification," *J Spacecr Rockets*, vol. 53, no. 5, 2016, doi: 10.2514/1.A33482.

[79] M. Mutyalarao and R. K. Sharma, "Optimal reentry time estimation of an upper stage from geostationary transfer orbit," *J Spacecr Rockets*, vol. 47, no. 4, 2010, doi: 10.2514/1.44147.

[80] C. Pardini and L. Anselmo, "Impact of the time span selected to calibrate the ballistic parameter on spacecraft re-entry predictions," *Advances in Space Research*, vol. 41, no. 7, 2008, doi: 10.1016/j.asr.2007.11.013.

[81] Y. Ding *et al.*, "Ballistic Coefficient Calculation Based on Optical Angle Measurements of Space Debris," *Sensors*, vol. 23, no. 18, 2023, doi: 10.3390/s23187668.

[82] A. K. Anilkumar, M. R. Ananthasayanam, and P. V Subba Rao, “A constant gain Kalman filter approach for the prediction of re-entry of risk objects,” *Acta Astronaut*, vol. 61, no. 10, 2007, doi: 10.1016/j.actaastro.2007.01.063.

[83] V. Ray, “Advances in Atmospheric Drag Force Modeling for Satellite Orbit Prediction and Density Estimation,” University of Colorado at Boulder, 2021.

[84] R. Jesionowski and P. Zarchan, “Comparison of Filtering Options For Ballistic Coefficient Estimation.”

[85] M. F. Storz, B. R. Bowman, J. I. Branson, S. J. Casali, and W. K. Tobiska, “High accuracy satellite drag model (HASDM),” *Advances in Space Research*, vol. 36, no. 12, 2005, doi: 10.1016/j.asr.2004.02.020.

[86] O. Jung, J. Seong, Y. Jung, and H. Bang, “Recurrent neural network model to predict re-entry trajectories of uncontrolled space objects,” *Advances in Space Research*, vol. 68, no. 6, pp. 2515–2529, Sep. 2021, doi: 10.1016/j.asr.2021.04.041.

[87] S. Paul and P. M. Mehta, “SATELLITE DRAG COEFFICIENT MODELING AND ORBIT UNCERTAINTY QUANTIFICATION USING STOCHASTIC MACHINE LEARNING TECHNIQUES How to become a successful engineering faculty View project.” [Online]. Available: https://www.researchgate.net/publication/353954605

[88] M. Howard, B. Klem, and J. Gorman, “RSO Characterization with Photometric Data Using Machine Learning,” 2015.

[89] R. Furfaro, R. Linares, and V. Reddy, “Space Debris Identification and Characterization via Deep Meta-Learning,” in *1st International Orbital Debris Conference*, 2019.

[90] E. Kerr *et al.*, “Light Curves for GEO object characterisation,” in *8th European Conference on Space Debris*, 2021.

[91] E. Kerr, E. Petersen, P. Talon, D. Petit, C. Dorn, and S. Eves, “Using AI to process light curves for GEO object characterisation,” in *Proceedings of the International Astronautical Congress, IAC*, 2021.

[92] M. Vasile, L. Walker, A. Campbell, V. Savitski, S. Marshall, and P. Murray, “Hyperspectral Classification of Space Objects,” in *Proceedings of the International Astronautical Congress, IAC*, 2022.

[93] M. Vasile *et al.*, “Intelligent characterisation of space objects with hyperspectral imaging,” *Acta Astronaut*, vol. 203, 2023, doi: 10.1016/j.actaastro.2022.11.039.

[94] C. R. V. L. C. D. A. H. Gregory P. Badura, “Recurrent Neural Network Autoencoders for Spin Stability Classification of Irregularly Sampled Light Curves,” in *The Advanced Maui Optical and Space Surveillance Technologies Conference*, 2022.

[95] R. Qashoa and R. Lee, “Classification of Low Earth Orbit (LEO) Resident Space Objects’ (RSO) Light Curves Using a Support Vector Machine (SVM) and Long Short-Term Memory (LSTM),” *Sensors*, vol. 23, no. 14, 2023, doi: 10.3390/s23146539.

[96] C. D. Enriquez and D. Amato, “Application of LSTMs to the Light Curve Inversion Problem ,” in *33rd AAS/AIAA Space Flight Mechanics Meeting*, 2023.

[97] H. Peng and X. Bai, “Recovering area-to-mass ratio of resident space objects through data mining,” *Acta Astronaut*, vol. 142, pp. 75–86, Jan. 2018, doi: 10.1016/j.actaastro.2017.09.030.

[98] T. S. Kelso and A. Sedona, “Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides Paper AAS 07-127 17 th AAS/AIAA Space Flight Mechanics Conference AAS 07-127 VALIDATION OF SGP4 AND IS-GPS-200D AGAINST GPS PRECISION EPHEMERIDES,” 2007.

[99] J. Rivera and X. Bai, “Improving the Orbit Propagation Accuracy of Two-Line-Element Satellite,” in *67th International Astronautical Congress*, Guadalajara, Mexico, 2016, pp. 26–30.

[100] N. Hatten and R. P. Russell, “A smooth and robust Harris-Priester atmospheric density model for low Earth orbit applications,” *Advances in Space Research*, vol. 59, no. 2, 2017, doi: 10.1016/j.asr.2016.10.015.

[101] V. Bernstein and M. Pilinski, “Drag Coefficient Constraints for Space Weather Observations in the Upper Thermosphere,” *Space Weather*, vol. 20, no. 5, 2022, doi: 10.1029/2021SW002977.

[102] V. Bernstein, M. Pilinski, and D. Knipp, “Evidence for drag coefficient modeling errors near and above the oxygen-to-helium transition,” *J Spacecr Rockets*, vol. 57, no. 6, 2020, doi: 10.2514/1.A34740.

[103] P. M. Mehta, C. A. McLaughlin, and E. K. Sutton, “Drag coefficient modeling for grace using Direct Simulation Monte Carlo,” *Advances in Space Research*, vol. 52, no. 12, 2013, doi: 10.1016/j.asr.2013.08.033.

[104] S. N. Paul, P. L. Sheridan, R. J. Licata, and P. M. Mehta, “Stochastic modeling of physical drag coefficient – Its impact on orbit prediction and space traffic management,” *Advances in Space Research*, vol. 72, no. 4, 2023, doi: 10.1016/j.asr.2023.06.006.

[105] P. M. Mehta, A. Walker, E. Lawrence, R. Linares, D. Higdon, and J. Koller, “Modeling satellite drag coefficients with response surfaces,” *Advances in Space Research*, vol. 54, no. 8, 2014, doi: 10.1016/j.asr.2014.06.033.

[106] P. M. Mehta *et al.*, “Satellite drag coefficient modeling for thermosphere science and mission operations,” *Advances in Space Research*, vol. 72, no. 12, 2023, doi: 10.1016/j.asr.2022.05.064.

[107] T. Hastie, R. Tibshirani, and J. Friedman, *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction*. 2001.

[108] L. Anselmo and C. Pardini, “Analysis of the Consequences in Low Earth Orbit of the collision between Cosmos 2251 and Iridium 33,” in *Proceedings of the 21st International Symposium on Space Flight Dynamics Earth Orbit*, 2009.

[109] L. Olivieri, P. A. Smocovich, C. Giacomuzzo, and A. Francesconi, “Characterization of the fragments generated by a Picosatellite impact experiment,” *Int J Impact Eng*, vol. 168, 2022, doi: 10.1016/j.ijimpeng.2022.104313.

[110] V. Ray, D. J. Scheeres, S. G. Hesar, and M. Duncan, “A Drag Coefficient Modeling Approach Using Spatial and Temporal Fourier Expansions for Orbit Determination,” *Journal of the Astronautical Sciences*, vol. 67, no. 3, 2020, doi: 10.1007/s40295-019-00200-4.

[111] N. K. Pavlis, S. A. Holmes, S. C. Kenyon, and J. K. Factor, “The development and evaluation of the Earth Gravitational Model 2008 (EGM2008),” *J Geophys Res Solid Earth*, vol. 117, no. 4, Apr. 2012, doi: 10.1029/2011JB008916.

[112] B. R. Bowman, W. K. Tobiska, F. A. Marcos, C. Y. Huang, C. S. Lin, and W. J. Burke, “A New Empirical Thermospheric Density Model JB2008 Using New Solar and Geomagnetic Indices,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008.

[113] D. A. Vallado and W. D. McClain, *Fundamentals of Astrodynamics and Applications*, 2nd Edition. Space Technology Library, 2013.

[114] “Space-Track.Org.” Accessed: Mar. 24, 2022. [Online]. Available: https://www.space-track.org/auth/login

[115] S. Yadav and S. Shukla, “Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification,” in *Proceedings - 6th International Advanced Computing Conference, IACC 2016*, 2016. doi: 10.1109/IACC.2016.25.

[116] S. J. Casali and W. N. Barker, “Dynamic calibration atmosphere (DCA) for the high accuracy satellite drag model (HASDM),” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2002. doi: 10.2514/6.2002-4888.

[117] M. D. Hejduk and D. E. Snow, “The Effect of Neutral Density Estimation Errors on Satellite Conjunction Serious Event Rates,” *Space Weather*, vol. 16, no. 7, 2018, doi: 10.1029/2017SW001720.

[118] “Regression Learner App - MATLAB & Simulink - MathWorks Italia.” Accessed: Feb. 18, 2022. [Online]. Available: https://it.mathworks.com/help/stats/regression-learner-app.html

[119] M. V Narkhede, P. P. Bartakke, and M. S. Sutaone, “A review on weight initialization strategies for neural networks,” *Artif Intell Rev*, vol. 55, no. 1, 2022, doi: 10.1007/s10462-021-10033-z.

[120] A. D. Gavrilov, A. Jordache, M. Vasdani, and J. Deng, “Preventing Model Overfitting and Underfitting in Convolutional Neural Networks,” *International Journal of Software Science and Computational Intelligence*, vol. 10, no. 4, 2019, doi: 10.4018/ijssci.2018100102.

[121] G. Isoletta, R. Opromolla, and G. Fasano, “Uncertainty-aware Cube algorithm for medium-term collision risk assessment,” *Advances in Space Research*, vol. 71, no. 1, 2023, doi: 10.1016/j.asr.2022.09.017.

[122] C. F. C. Macêdo and E. M. Rocco, “Evaluation of orbital decay of a satellite at low altitude due to atmospheric drag as a function of solar activity,” in *Journal of Physics: Conference Series*, 2019. doi: 10.1088/1742-6596/1365/1/012027.

[123] T. Lechtenberg, C. A. McLaughlin, T. Locke, and D. M. Krishna, “Thermospheric density variations: Observability using precision satellite orbits and effects on orbit propagation,” *Space Weather*, vol. 11, no. 1, 2013, doi: 10.1029/2012SW000848.

[124] J. T. Emmert, H. P. Warren, A. M. Segerman, J. M. Byers, and J. M. Picone, “Propagation of atmospheric density errors to satellite orbits,” *Advances in Space Research*, vol. 59, no. 1, 2017, doi: 10.1016/j.asr.2016.07.036.

[125] “Lemur-2 - Gunter’s Space Page.” Accessed: Apr. 04, 2022. [Online]. Available: https://space.skyrocket.de/doc_sdat/lemur-2.htm

[126] “LEMUR-2-ANGELA / 42752 / 2017-019B - Satellite Orbit Data.” Accessed: Apr. 04, 2022. [Online]. Available: https://orbit.ing-now.com/satellite/42752/2017-019b/lemur-2-angela/

[127] “LEMUR-2-ZUPANSKI / 42752 / 2017-019B - Satellite Orbit Data.” Accessed: Apr. 04, 2022. [Online]. Available: https://orbit.ing-now.com/satellite/42752/2017-019b/lemur-2-zupanski/

[128] “Lemur-2 - Satellite Missions - eoPortal Directory.” Accessed: Apr. 04, 2022. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/l/lemur

[129] M. Di Carlo, S. da Graça Marto, and M. Vasile, “Extended analytical formulae for the perturbed Keplerian motion under low-thrust acceleration and orbital perturbations,” *Celest Mech Dyn Astron*, vol. 133, no. 3, 2021, doi: 10.1007/s10569-021-10007-x.

[130] D. Amato, C. Bombardelli, G. Baù, V. Morand, and A. J. Rosengren, “Non-averaged regularized formulations as an alternative to semi-analytical orbit propagation methods,” *Celest Mech Dyn Astron*, vol. 131, no. 5, 2019, doi: 10.1007/s10569-019-9897-1.

[131] G. Baù, C. Bombardelli, J. Pelàez, and E. Lorenzini, “Non-singular orbital elements for special perturbations in the two-body problem,” *Mon Not R Astron Soc*, vol. 454, no. 3, 2015, doi: 10.1093/mnras/stv2106.

[132] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. 2019. doi: 10.1017/9781108380690.

[133] Bate RR, Mueller DD, White JE, and Saylor WW, *Fundamentals of Astrodynamics*. Courier Dover Publications, 2020.

[134] O. Montenbruck, E. Gill, and F. Lutze, “ Satellite Orbits: Models, Methods, and Applications ,” *Appl Mech Rev*, vol. 55, no. 2, 2002, doi: 10.1115/1.1451162.

[135] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin, “NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues,” *J Geophys Res Space Phys*, vol. 107, no. A12, 2002, doi: 10.1029/2002JA009430.

[136] U. Walter, *Astronautics: The physics of space flight*. 2019.

[137] “SatCen - The Recognized Space Picture (RSP) Display study.” Accessed: Jan. 18, 2023. [Online]. Available: https://www.satcen.europa.eu/page/the-recognized-space-picture-rsp-display-study

[138] L. Leveque, “Space Situational Awareness and Recognized Picture,” in *Handbook of Space Security: Policies, Applications and Programs*, P. L. and R. J. and M. D. and G. C. Schrogl Kai-Uwe and Hays, Ed., New York, NY: Springer New York, 2015, pp. 699–715. doi: 10.1007/978-1-4614-2029-3_46.

[139] “SPICE Toolkit.” [Online]. Available: https://naif.jpl.nasa.gov/naif/toolkit.html

[140] J. A. Slater and S. Malys, “WGS 84 — Past, Present and Future,” 1998. doi: 10.1007/978-3-662-03714-0_1.

[141] V. Pesce, A. Colagrossi, and S. Silvestrini, *Modern Spacecraft Guidance, Navigation, and Control: From System Modeling to AI and Innovative Applications*. 2022. doi: 10.1016/C2020-0-03563-2.

[142] D. Vallado and W. McClain, “Fundamentals of astrodynamics and applications, 2004,” *Space Technology Library, Microcosm Press & Kluwer Academic Publishers*, 2007.

[143] C. N. McGrath, R. A. Clark, and M. Macdonald, “Novel concept of satellite manoeuvre planning using graph theoretical techniques,” *Advances in Space Research*, vol. 67, no. 11, pp. 3775–3784, Jun. 2021, doi: 10.1016/j.asr.2020.06.008.

[144] J. Sheng and Y. Geng, “Normalized nodal distance-based J2-perturbed ground-track adjustment with apsidal-altitude constraints,” *Aerosp Sci Technol*, vol. 140, Sep. 2023, doi: 10.1016/j.ast.2023.108418.

[145] T. Li, J. Xiang, Z. Wang, and Y. Zhang, “Circular revisit orbits design for responsive mission over a single target,” *Acta Astronaut*, vol. 127, pp. 219–225, Oct. 2016, doi: 10.1016/j.actaastro.2016.05.037.

[146] O. Abdelkhalik, “Initial Orbit Design from Ground Track Points,” *J Spacecr Rockets*, vol. 47, no. 1, pp. 202–205, 2010, doi: 10.2514/1.44857.

[147] O. Abdelkhalik and D. Mortari, “Orbit design for ground surveillance using genetic algorithms,” *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1231–1235, 2006, doi: 10.2514/1.16722.

[148] X. Lin, G. Zhang, and H. Zhang, “Multi-target ground-track adjustment with a single coplanar impulse,” *Aerosp Sci Technol*, vol. 119, Dec. 2021, doi: 10.1016/j.ast.2021.107135.

[149] T. C. Co, C. Zagaris, and J. T. Black, “Responsive satellites through ground track manipulation using existing technology,” in *Journal of Spacecraft and Rockets*, American Institute of Aeronautics and Astronautics Inc., 2013, pp. 206–216. doi: 10.2514/1.A32263.

[150] G. Zhang and X. Cao, “Coplanar ground-track adjustment using time difference,” *Aerosp Sci Technol*, vol. 48, pp. 21–27, Jan. 2016, doi: 10.1016/j.ast.2015.10.023.

[151] G. Zhang, X. Cao, and D. Mortari, “Analytical approximate solutions to ground track adjustment for responsive space,” *IEEE Trans Aerosp Electron Syst*, vol. 52, no. 3, pp. 1366–1383, Jun. 2016, doi: 10.1109/TAES.2016.140644.

[152] G. Zhang and J. Sheng, “Impulsive ground-track adjustment for assigned final orbit,” *J Spacecr Rockets*, vol. 53, no. 4, pp. 599–609, 2016, doi: 10.2514/1.A33447.

[153] T. C. Co and J. T. Black, “Responsiveness in low orbits using electric propulsion,” in *Journal of Spacecraft and Rockets*, American Institute of Aeronautics and Astronautics Inc., 2014, pp. 938–945. doi: 10.2514/1.A32405.

[154] M. M. Guelman and A. Shiryaev, “Closed-loop control of earth observation satellites,” *J Spacecr Rockets*, vol. 56, no. 1, pp. 82–90, 2019, doi: 10.2514/1.A34134.

[155] S. H. Mok and H. Bang, “Optimal Multi-Target Overflight Using Ground-Track Adjustment,” *Journal of the Astronautical Sciences*, vol. 68, no. 1, pp. 150–171, Mar. 2021, doi: 10.1007/s40295-020-00240-1.

[156] X. Yu, K. Liu, and Q. Chen, “Constellation design for responsive visiting based on ground track adjustment,” *Proc Inst Mech Eng G J Aerosp Eng*, vol. 232, no. 12, pp. 2300–2316, Sep. 2018, doi: 10.1177/0954410017716479.

[157] O. L. de Weck, U. Scialom, and A. Siddiqi, “Optimal reconfiguration of satellite constellations with the auction algorithm,” *Acta Astronaut*, vol. 62, no. 2–3, pp. 112–130, Jan. 2008, doi: 10.1016/j.actaastro.2007.02.008.

[158] C. N. McGrath and M. Macdonald, “General perturbation method for satellite constellation reconfiguration using low-thrust maneuvers,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1676–1692, 2019, doi: 10.2514/1.G003739.

[159] D. H. Maling, “Coordinate systems and map projections for GIS,” *Geographical information systems. Vol. 1: principles*, 1991.

[160] O. Montenbruck and T. Pfleger, *Astronomy on the Personal Computer*. 2000. doi: 10.1007/978-3-642-03436-7.

[161] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” *Reliab Eng Syst Saf*, vol. 91, no. 9, pp. 992–1007, Sep. 2006, doi: 10.1016/j.ress.2005.11.018.

[162] K. Deb, “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction,” in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, Springer London, 2011, pp. 3–34. doi: 10.1007/978-0-85729-652-8_1.

[163] N. Gunantara, “A review of multi-objective optimization: Methods and its applications,” *Cogent Eng*, vol. 5, no. 1, pp. 1–16, Jan. 2018, doi: 10.1080/23311916.2018.1502242.

[164] K. Deb, “Multi-objective optimization,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Second Edition*, Springer US, 2014, pp. 403–450. doi: 10.1007/978-1-4614-6940-7_15.

[165] C. H. Yeh, “A Problem-based Selection of Multi-attribute Decision-making Methods,” *International Transactions in Operational Research*, vol. 9, no. 2, 2002, doi: 10.1111/1475-3995.00348.

[166] J. Sang, J. C. Bennett, and C. H. Smith, “Estimation of ballistic coefficients of low altitude debris objects from historical two line elements,” *Advances in Space Research*, vol. 52, no. 1, 2013, doi: 10.1016/j.asr.2013.03.010.

[167] “CelesTrak: Current GP Element Sets.” [Online]. Available: https://celestrak.org/NORAD/elements/

[168] “Deimos-1/GEOSAT-1 - eoPortal.” [Online]. Available: https://www.eoportal.org/satellite-missions/deimos-1

[169] “UK-DMC-2 (United Kingdom - Disaster Monitoring Constellation-2) - eoPortal.” [Online]. Available: https://www.eoportal.org/satellite-missions/uk-dmc-2

[170] “RISAT-1 (Radar Imaging Satellite-1) - eoPortal.” [Online]. Available: https://www.eoportal.org/satellite-missions/risat-1

[171] “HJ-1(Huan Jing-1: Environmental Protection & Disaster Monitoring Constellation) - eoPortal.” [Online]. Available: https://www.eoportal.org/satellite-missions/hj-1

[172] R. J. Licata, P. M. Mehta, W. K. Tobiska, and S. Huzurbazar, “Machine-Learned HASDM Thermospheric Mass Density Model With Uncertainty Quantification,” *Space Weather*, vol. 20, no. 4, 2022, doi: 10.1029/2021SW002915.

[173] S. Bruinsma *et al.*, “Thermosphere and satellite drag,” *Advances in Space Research*, 2023, doi: 10.1016/J.ASR.2023.05.011.

[174] M. Manzi and M. Vasile, “Discovering Unmodeled Components in Astrodynamics with Symbolic Regression,” in *2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings*, 2020. doi: 10.1109/CEC48606.2020.9185534.

[175] M. Manzi and M. Vasile, “Autoencoder-based Thermospheric Density Model for Uncertainty Quantification and Real-Time Calibration,” in *8th European Conference on Space Debris*, 2021.

[176] P. L. McDermott and C. K. Wikle, “Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data,” *Entropy*, vol. 21, no. 2, 2019, doi: 10.3390/e21020184.

List of publications

Journal papers

- Cimmino, N., Opromolla, R., Fasano, G. (2023). Machine learning-based approach for ballistic coefficient estimation of resident space objects in LEO. *Advances in Space Research* 71.12: 5007-5025. (https://doi.org/10.1016/j.asr.2023.02.007)
- Cimmino, N., Isoletta, G., Opromolla, R., Fasano, G., Basile, A., Romano, A., Peroni, M., Panico, A., Cecchini, A. (2021). Tuning of NASA Standard Breakup Model for fragmentation events modelling. *Aerospace*, 8(7), 185. (https://doi.org/10.3390/aerospace8070185)
- Cimmino, N., Amato, D., Opromolla, R., Fasano, G. (2024). Recurrent Neural Networks for ballistic coefficient estimation of resident space objects in Low Earth Orbit. *Ready for submission*.
- Argirò, A., Cimmino, N., Opromolla, R., Fasano, G. (2024). A decision-making architecture for responsive manoeuvring optimization of multiple Earth observation satellites. *Ready for submission*.

Conference papers

- Cimmino, N., Isoletta, G., Opromolla, R., Fasano, G., Amato, D. Improving Ballistic Coefficient Estimation of Resident Space Objects in Low Earth Orbit. Proceedings of *74th International Astronautical Congress*, Baku, Azerbaijan, 2-6 October 2023 (**speaker**)
- Cimmino, N., Argirò, A., Isoletta, G., Opromolla, R., Fasano. G. (2023). Optimized multi-satellite manoeuvring strategy for responsive Earth observation. Proceedings of *The Aerospace Europe Conference*, Lausanne, Switzerland, 9-13 July 2023
- Cimmino, N., Isoletta, G., Opromolla, R., Fasano, G., Basile, A., Pesacane, O., Romano, A., Rigamonti, M., Peroni, M., Panico, A., Cecchini, A. Earth orbiting resident space objects characterization based on astrometric data. Proceedings of *IEEE International Workshop on Metrology for Aerospace*, Milan, Italy, 19-21 June 2023 (10.1109/MetroAeroSpace57412.2023.10189955)
- Cimmino, N., Isoletta, G., Opromolla, R., Fasano, G. Performance and sensitivity analysis of machine learning-based approaches for resident space object characterization. Proceedings of *73rd International Astronautical Congress*, Paris, France, 18-22 September 2022. (**speaker**)
- Isoletta, G., Cimmino, N., Opromolla, R., Fasano, G. Uncertainty evaluation tool for medium-term Low Earth Orbit propagation. Proceedings of *73rd International Astronautical Congress*, Paris, France, 18-22 September 2022

- Isoletta, G., Cimmino, N., Opromolla, R., Fasano, G., Romano, A., Basile, A., Pisacane, O., Peroni, M., Panico, A., Cecchini, A. Orbital propagation challenges and solutions for SST fragmentation services. Proceedings of *72nd International Astronautical Congress*, Dubai, United Arab Emirates, 25-29 October 2021